



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH EDITORU TYPOVÝCH ŠTÍTKŮ

DESIGN OF NAMEPLATES EDITOR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Svoboda

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Pavel Houška, Ph.D.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav automatizace a informatiky
Student: **Bc. Filip Svoboda**
Studijní program: Strojní inženýrství
Studijní obor: Aplikovaná informatika a řízení
Vedoucí práce: **Ing. Pavel Houška, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh editoru typových štítků

Stručná charakteristika problematiky úkolu:

Současná výroba vyžaduje přesné značení a identifikaci všech výrobků, které opouští výrobu. Podle povahy výrobku a objemu výroby se používá několik technologií od tisku a lepení typových štítků, po tampónový tisk a laserové značení. Obsah typového štítku se během času mění a tyto změny je nutné provádět se šablonou štítku, která se vytváří pro konkrétní tiskárnu. Tato práce se zabývá problematikou malosériové výroby, kde se z mnoha důvodů používá tisk na samolepící štítky. Výsledkem práce má být editor štítků, který bude součástí software výrobních a montážních linek.

Cíle diplomové práce:

Seznamte se s problematikou typových štítků a tiskáren typových štítků;
Analyzujte požadavky na obsah a rozložení štítků s ohledem na možnosti dostupných tiskových technologií;
Navrhněte a realizujte editaci a ukládání typových štítků;
Navrhněte a realizujte export štítků do formátu ZPL 2;
Realizovaný editor ověřte a zhodnoťte.

Seznam doporučené literatury:

KATO Hiroko, TAN Keng T. a CHAI Douglas. Barcodes for mobile devices. Cambridge ; New York: Cambridge University Press, 2010, viii, 257 s., [19] s. příl. : il. (některé bar.) ; 26 cm. ISBN 978-0-5-1-88839-4.

WOOD Larry E. User interface design. Boston: CRC Press, 1998, 312 s. ISBN 0-8493-3125-0.

ZEBRA. Programming Guide for ZPL II ZBI 2 Set-Get-Do Mirror WML. Lincolnshire, 2016. Dostupné také z: <https://www.zebra.com>.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

doc. Ing. Radomil Matoušek, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Diplomová práce se zabývá problematikou typových štítků, používaných pro značení výrobků v sériové výrobě. V rešeršní části práce je uvedena problematika používaných médií typových štítků a jejich tisku. Je analyzována problematika obsahu a návrhu štítků. Výsledkem práce je grafický editor pro návrh štítků a jejich převod do jazyku ZPL II. Tento jazyk používají zejména tiskárny firmy Zebra Technologies, které jsou velmi rozšířené.

ABSTRACT

The diploma thesis deals with the problematics of type labels, used for marking of products in series production. The research part of the thesis deals with the issues of the used label media and label printing. The issue of content and design of labels is analyzed. The result of thesis is a graphic editor for designing labels and their translation into ZPL II. This language is mainly used by Zebra Technologies printers, which are very common used in industry.

KLÍČOVÁ SLOVA

Typové štítky, návrh editoru, LabVIEW, .NET Framework, termotisk.

KEYWORDS

Type plates, editor design, LabVIEW, .NET Framework, thermoprint.

BIBLIOGRAFICKÁ CITACE

SVOBODA, F. *Návrh editoru typových štítků*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 66 s. Vedoucí diplomové práce Ing. Pavel Houška, Ph.D..

PODĚKOVÁNÍ

Děkuji panu Ing Pavlu Houškovi Ph.D. za vedení diplomové práce a odborné rady. Dále děkuji své rodině za podporu během studia.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Pavla Houšky, Ph.D. a s použitím literatury uvedené v seznamu literatury.

V Brně dne 26. 5. 2017

.....

Filip Svoboda

OBSAH

1	ÚVOD.....	15
2	OBSAH TYPOVÝCH ŠTÍTKŮ	17
2.1	Textové údaje	17
2.2	Čárové kódy.....	18
2.3	Dvourozměrné kódy	20
2.4	Obrazová data	24
2.5	Požadavky průmyslu na vlastnosti a obsah štítků	24
3	TISK TYPOVÝCH ŠTÍTKŮ	25
3.1	Média pro tisk štítků	25
3.2	Sítotisk	26
3.3	Inkoustový tisk	26
3.3.1	Souvislý proud kapek	26
3.3.2	Kapky při pulzu	27
3.4	Tamponový tisk	29
3.5	Termotisk.....	29
3.5.1	Přímý termotisk	29
3.5.2	Termotransferový tisk.....	30
3.6	Laserové značení	31
3.6.1	Znační plastů.....	31
3.6.2	Značení kovů	32
3.6.3	Značení elektronických součástek	32
3.7	Ostatní mechanické značení	32
4	ZEBRA TECHNOLOGIES	33
4.1	Historie	33
4.2	ZPL II	34
4.2.1	Základní struktura.....	34
4.2.2	Tisknutelný obsah.....	34
4.3	ZebraDesigner	35
4.3.1	Výhody softwaru	35
4.3.2	Nevýhody softwaru.....	35
4.3.3	Práce s programem	36
5	NÁVRH EDITORU TYPOVÝCH ŠTÍTKŮ	37
5.1	Koncepce editoru typových štítků	37
5.2	Základní rozbor ergonomie	40
6	REALIZACE EDITORU TYPOVÝCH ŠTÍTKŮ	43
6.1	Obsahová část	44
6.1.1	Prototype.....	45
6.1.2	Text.....	46
6.1.3	Čárové kódy.....	47
6.1.4	Dvoudimenzionální kódy	49
6.1.5	Obrazová data	50
6.2	Část Editor	52
6.3	Preprocesor a překladač do ZPL II	54
6.4	Vlastnosti realizovaného editoru	55

6.5	Další vývoj editoru	56
7	ZÁVĚR.....	57
8	SEZNAM POUŽITÉ LITERATURY	59
9	SEZNAM OBRÁZKŮ.....	61
10	SEZNAM PŘÍLOH	63

1 ÚVOD

Moderní řízení výroby vyžaduje jednoznačnou identifikaci výrobků, která se řeší různými typy identifikačních štítků. Ve velkosériové výrobě převládá použití značení pomocí laseru při vytváření těchto štítků. V malosériové výrobě se častěji používá termotisk na tisk samolepek.

Takto označené produkty lze poté snadno dohledat v případě reklamací a změn. Lze tak zjistit poruchovost nejen samotného výrobku, ale i poruchovost série a další důležitá data.

Pro strojové načítání identifikace součástí slouží čárové a dvoudimenzionální kódy, které mohou nést i rozšiřující informace o součásti. V některých výrobních postupech nelze bez naskenování typového štítku pokračovat v montáži. Při takovém použití je poté možné dohledat konkrétní díl téměř v jakékoliv sestavě a v případě problémů může být nahrazen.

Technologie RFID (Radio Frequency Identification) štítků zatím převládá zejména pro označení přípravků, forem a podobně. Použití pro značení výrobků limituje vyšší cena RFID štítku v porovnání s ostatními technologiemi.

Se vzrůstajícím počtem a složitostí štítků vzrůstá i složitost jejich návrhu. Při použití tiskového média ve formě samolepky je nejrozšířenější metodou tisku termotisk v obou svých variantách. V nabídce termotiskáren existuje několik firem, přičemž tato práce je zaměřena na využití zejména produktů firmy Zebra Technologies, které jsou velmi často používány ve výrobních provozech v okolí Brna. Tato firma stojí i za vývojem jazyka pro návrh štítků, který se jmenuje ZPL.

Z důvodu několika problémů při ručním návrhu štítků bylo rozhodnuto navrhnout editor, který by měl některé tyto problémy vyřešit. Použitelnost původních štítků byla závislá na konkrétní tiskárně a parametrech jejích součástí. V případě poruchy tiskárny nebylo možné, bez znalosti písma použitého při návrhu štítku, tisknout tento štítek v navrhnutém vzhledu na jiné tiskárně.

Navrhovaný editor by měl sloužit pro tvorbu typových štítků a jejich uložení do souboru, který by byl při dodržení několika drobných podmínek přenositelný. To znamená, že v případě poruchy tiskárny by byla možná její náhrada bez větších problémů.

Navrhnuté štítky lze pak tisknout i bez přítomnosti samotného editoru pomocí překladového nástroje, který daný štítek dokáže přeložit do ZPL II. Tedy do podoby obrázku, který je následně přeložen do ZPL II. Tento mezikrok je nutný pro garantování výsledku při tisku a odstranění problémů s volbou písma.

2 OBSAH TYPOVÝCH ŠTÍTKŮ

Typové štítky se používají pro jednoznačnou identifikaci montovaného dílce či sestavy dílců. Tyto informace slouží většinou k internímu použití ve firmě. Mohou ale obsahovat i informace pro spotřebitele. Často se uvádí i jméno a logo firmy, případně i název a logo součásti nebo série. Obsah typových štítků není ukotven v žádné literatuře, kromě interních dokumentů jednotlivých firem. Neexistuje tedy norma či směrnice, která by seskupovala grafický vzhled štítků.

Existuje pouze několik obecných zásad, které se dodržují téměř mimovolně. Například se vždy volí kontrastní barvy pro podklad a data štítku. Většina typových štítků je pouze jednobarevná. Barva podkladu i obsahu je závislá na vybrané technologii tisku.

Moderním obsahem štítku se stává určitá forma elektronického mikročipu, ve kterém jsou uloženy informace o součásti. Čtení pak probíhá pomocí bezdrátové komunikace. Štítky využívající tuto technologii lze i přeprogramovat a v průběhu výrobního procesu aktualizovat obsažená data. [1]

2.1 Textové údaje

Tyto údaje se řadí mezi nejběžnější. První typové štítky obsahovaly pouze textové informace, protože jiný způsob zápisu informací neexistoval nebo se nepoužíval. Jde o různé formy pořadových čísel, identifikátorů a dalších informací. Do textových údajů také spadá název vyráběné části, případně datum výroby kvůli identifikaci. Tyto texty obvykle slouží k internímu firemnímu zařazení během reklamací. Přes tyto údaje lze většinou dohledat výrobní linku, čas výroby i operátora zodpovědného za výrobu v daný čas. Na Obr. 1 je viditelná plocha pro výrobní číslo a pro výkon motoru na historickém štítku.



Obr. 1: Historický typový štítek [2]

Fonty a velikost

Při použití textových údajů se musí zvažovat také úskalí použití různých fontů a jejich velikostí. Obě tyto vlastnosti jsou ovlivněny použitou tiskárnou. Některé tiskárny nejsou schopny kvůli svému nízkému rozlišení tisknout čitelně malé písmo. Další tiskárny mohou mít pouze omezenou paletu fontů, které jsou použitelné. Při používání např. češtiny hrozí problémy se zobrazením diakritiky.

2.2 Čárové kódy

Kódy jsou sestaveny ze střídajících se bílých a černých pruhů různých šířek. Kódů existuje velké množství druhů a mutací. Pro větší přehlednost se může uvádět pod čárovým kódem i obsah ve formě klasického textu, což umožňuje čitelnost i pro člověka.

UPC

Jedná se o universální kód výrobků (universal product code), který se zaváděl v supermarketech od roku 1973. Je základním kamenem pro rozšíření a mutace. Jedním z těchto rozšíření je EAN-13. Byl navrhován s myšlenkou, aby bylo možné jednoznačně určit výrobce a produkt.

Dokáže zakódovat pouze čísla a vždy má předem danou délku, která odpovídá použité mutaci kódu. Těchto mutací je pět a jsou označovány písmeny A-E. Ovšem kromě skupiny A a E se jiné mutace využívají velmi zřídka. [3]

EAN-13

Tento typ čárového kód je v dnešní době nejrozšířenější zejména na území Evropy. Objevuje se téměř na všech potravinách a zboží, kde slouží k načtení do pokladního systému. Obsahuje dva datové bloky po šesti znacích (číslech). Sekce jsou od sebe odděleny středovou částí, tu lze vidět na Obr. 2.

Poslední třináctá číslice je výsledek kontrolního součtu. Každý znak je reprezentován sedmi moduly. Každý modul má konstantní šířku. Moduly se rozdělují mezi dvě mezery a dvě čáry, šířka se pohybuje mezi jedním a čtyřmi moduly. [3]

Existuje i unifikovaný systém, který definuje, že u výrobků jsou první tři číslice identifikační kód země (pro ČR 859), dalších čtyři až šest číslic je identifikátor přidělený firmě. [4]



Obr. 2: Ukázka EAN-13

UPC A

Kód se používá především v Severní Americe a Kanadě na označování produktů v obchodech. Obsahuje dvanáct pozic pro čísla. První číslo určuje druh výrobku. Je téměř shodný s EAN-13, rozdíl je pouze ve významu jednotlivých pozic. Kvůli možné ignoraci první pozice mohou čtečky EAN-13 vždy přečíst UPC A, ale ne vždy naopak. [3]

UPC E a EAN-8

Pokud jsou výrobky tak malé, že na ně nelze umístit plný dvanáctimístný kód, lze použít tuto mutaci, která zobrazuje pouze osm číslic. Také vypouští středový oddělovač a kontrolní součet. [3]

Code 128

Kód může obsahovat číslice ale oproti EAN-13 i znaky. Ty mohou být, jak malé z rozsahu a-z, tak velké z rozsahu A-Z. Mimo tyto znaky dokáže zakódovat i ostatní znaky, jak je vidět na Obr. 3. Čárový kód Code 128 potřebuje na zakódování určitého množství informace nejméně prostoru ze všech čárových kódů.

Kód je složen z počáteční prázdné (tiché) zóny, startovního znaku, zakódovaných dat a ukončovacího znaku, který je následován další ukončovací tichou zónou. Každý zobrazený znak obsahuje tři čáry a tři mezery. Každá z těchto částí může mít jeden až čtyři moduly. Celkově se znak skládá z jedenácti modulů. Existuje ale výjimka v podobě ukončovacího znaku, který je tvořen třinácti moduly. Dále také platí pravidlo, že čáry mají sudý počet modulů a mezery lichý počet. Díky tomu lze snadno kontrolovat proces čtení.

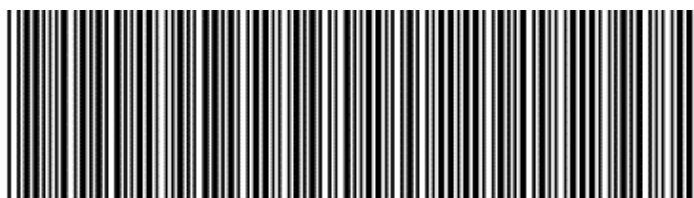
Dle kódovaných znaků se volí nejvhodnější ze tří abeced, kdy každá obsahuje jiné zastoupení znaků. Při kódování znaků, které neleží ve stejné abecedě lze přepínat mezi abecedami i v jednom kódu díky speciálním znakům. Data jsou chráněna proti porušení díky kontrolnímu součtu pracujícím s modulo 103. [3]



Obr. 3: Ukázka Code 128

Code39 a Code39 Mod 43

V roce 1974 byl tento kód vyvinut jako první, který mohl reprezentovat zároveň čísla i písmena. Kód má proměnlivou délku a každý znak znázorňuje pomocí čtyři mezer a pěti čar. Z těchto prvků jsou vždy tři široké a šest úzkých. Kód neobsahuje pouze malá písmena, ale obsahuje i znak hvězdičky. Tato se používá pro definování začátku a konce. Rozšířená verze obsahuje navíc kontrolní znak. Ukázka kódu bez kontrolního součtu je vidět na Obr. 4. [3]



Obr. 4: Ukázka Code 39

2.3 Dvourozměrné kódy

Kvůli potřebě vyšší datové hustoty, tedy pro zápis většího množství dat na danou plochu, bylo nutné rozšířit obyčejné čárové kódy o další rozměr. Čárové kódy jsou ve své podstatě nezávislé na své výšce. Pokud by byly vysoké pouze jeden tisknutelný bod a čtečka by byla schopná rozlišit tento řádek teček, bylo by to funkční zobrazení kódu. U čárových kódů jde tedy při volbě velikosti zejména o čitelnost a o ochranu kódu proti poškození.

Rozšířením zápisu do kolmého rozměru se zvýší množství kódovatelných dat, ale i dalších, která jsou potřebná pro detekci chyb a případné opravy nesené informace. Hlavními výhodami je tedy velká datová kapacita, vyšší odolnost proti chybám a rozšířené možnosti ukládání. [3]

Datová kapacita

Díky tomu, že se data kódují ve dvou směrech, je takový kód prostorově efektivnější. Ve srovnání například s typickým představitelem čárových kódů EAN-13, který dokáže kódovat třináct znaků, dokáže dvourozměrný QR kód zakódovat až 7089 znaků. Neboli QR dokáže zobrazit stejné množství dat jako EAN-13 na jedné desetině plochy. [5]

Odolnost proti chybám

Dvourozměrné kódy vynikají odolností proti poškození či jiné formě zhoršení čitelnosti například ušpiněním. Dosahují toho jednak tím, že data mohou být zapsána v jednom kódu několikrát ale také tím, že často obsahují nějakou formu korekce chyb. Příkladem takového opatření jsou Reed-Solomonovy kódy. Míra schopnosti restaurovat kódovaná data vychází z nastavení při generování kódu. [5]

Rozšířené možnosti ukládání

Další výhodou dvourozměrných kódů je schopnost ukládat libovolná data. Mimo obvykle používaných textových údajů lze zakódovat URL, adresy, kontakty a další. Tato výhoda se projevuje zejména při použití mobilních telefonů nebo tabletů jako čteček. Díky tomu

je možné na vizitce naskenovat kromě jména a příjmení třeba i koordináty sídla firmy a podobně. Tato výhoda ovšem přináší i větší nároky na velikost kódu. Kód, který obsahuje pouze telefonní číslo, bude menší než ten, který obsahuje veškeré údaje o firmě včetně odkazu na internetové stránky. [5]

Nevýhody dvoudimenzionálních kódů

Mimo výše zmíněné výhody mají dvoudimenzionální kódy i nevýhody. Ze své podstaty musí být orientované nějakým směrem. Proto musí obsahovat poziční značky, které umožňují čtení. Nejčastěji používaná lineární čtečka čárových kódů se pro čtení dvoudimenzionálních kódů nedá použít. Musí se použít kamera a poté analýza obrazu. S tím souvisí i delší doba čtení kódu, která je kromě jiného závislá na světelných podmínkách a také velikosti kódu.

Data Matrix

Kód, který vyvinula firma International Data Matrix Inc. v roce 1987. V dnešní době jsou nejčastěji používané dva podtypy a to ECC 000-140 a ECC 200. První verze osahuje nastavitelnou míru konvoluční opravy chyb. ECC 200 používá pro opravu chyb Reed-Solomonovy kódy.

V závislosti na použitém kódování je minimální velikost kódu pro ECC 000-140 9x9 buněk (bodů) a maximální velikost je 49x49 buněk. Oproti ECC 200 je rozměr vždy lichý počet buněk. Velikost ECC 200 je závislá na stupni ochrany dat a pohybuje se v rozmezí 10x10 až 144x144 buněk.

Data Matrix se skládá z bílých a černých bodů ve tvaru čtverce. Pro zjištění orientace a velikosti jsou dvě strany tvořeny plnými čarami ve tvaru písmene L. Zbylé dvě strany jsou tvořeny přerušovanými čarami, které určují počet sloupců a řádků, které definují oblast, kde je uložen zakódovaný znak. Díky plným stranám a přerušovaným stranám lze zjistit velikost kódu a jeho zkrácení, pokud není nasnímán v dokonale kolmém směru. Struktura je uvedena na Obr. 5. [3]



Obr. 5: Ukázka struktury Data Matrix [3]

Tvar kódu může být čtverec i obdélník. Kvůli zvýšení spolehlivosti a rychlosti čtení byla v ECC 200 přidána schopnost dělit data do bloků. Data jsou dělena do bloku, pokud by velikost kódu přesáhla 24x24 buněk. Nejmenším množstvím dat, které lze zapsat je šest číslic, tři znaky nebo jeden byte obecných dat. Maximální kapacita je pak 3116 čísel, 2335 znaků nebo 1556 bytů obecných dat. Data Matrix obsahuje pouze dvě znakové sady

a to 168 znaků odpovídajících ISO 646 nebo uživatelsky definovanou sadu o velikosti 256 znaků. U starší verze ECC 000-140 lze uživatelsky stanovit úroveň schopnosti opravy chyb, u novější verze ECC 200 nelze úroveň zvolit a je pevně daná velikostí kódu. [3]

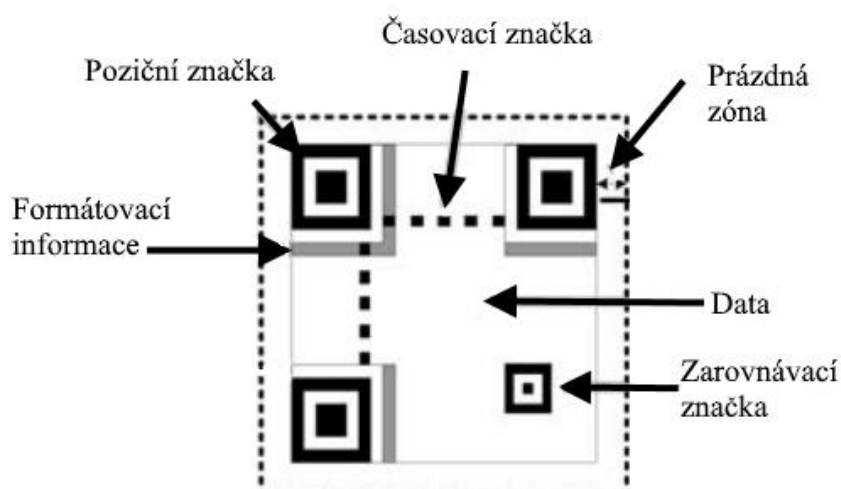
QR kód

QR (Quick Response) neboli kód pro rychlou odezvu, byl vyvinut firmou Denso Wave v roce 1994. Jedná se o dvoudimenzionální maticový kód. Jak samotný název napovídá, jedná se o kód primárně vyvinutý pro velmi rychlé snímání. Také je použitelný pro zakódování japonských znaků. Další nespornou výhodou je obrovská datová kapacita (menší než u Data Matrix na stejné ploše), datová hustota a volitelný stupeň korekce dat. Na základech QR kódu jsou postaveny některé další deriváty, jako je Mobile Code a MS-Code. [3]

Kód je tvořen z několika složek o různých funkcích. Jedná se o poziční a vyhledávací značky, které jsou ve tvaru překrývajících se čtverců se stejným středem. Tyto značky jsou umístěné ve třech rozích, které umožňují jak vyhledání kódu na ploše, tak i jeho natočení. Kolem kódu musí být volná (tichá) zóna. [3]

Další složkou jsou časovací značky, které jsou umístěné jako propojení tří pozičních značek. Tvoří je přerušované čáry. Jsou používány pro výpočet těžiště každé datové buňky, a případné korekce zkreslení. [3]

Následující složkou jsou zarovnávací značky, které se využívají k detekci lokálních zkreslení. Díky zjištění středu této značky a následné úpravě center datových buněk lze zkreslení korigovat. Černá tečka ve středu této značky se snadno identifikuje a tím je zajištěna rychlost korekce. Dále kód obsahuje informace o formátu uložených dat. Tyto data obsahují například verzi, úroveň korekce chyby a maskovací značku použitou v daném kódu. Tyto informace se čtou jako první. Nezbytnou součástí kódu jsou samotná uložená data. Jsou chráněna díky Reed-Solomonovým kódům. Data mohou obsahovat jak jednoduché údaje v podobě textu nebo čísel, tak i složitější obsah jako jsou webové odkazy, obrázky a další. Popis kódu je viditelný na Obr. 6. [3]



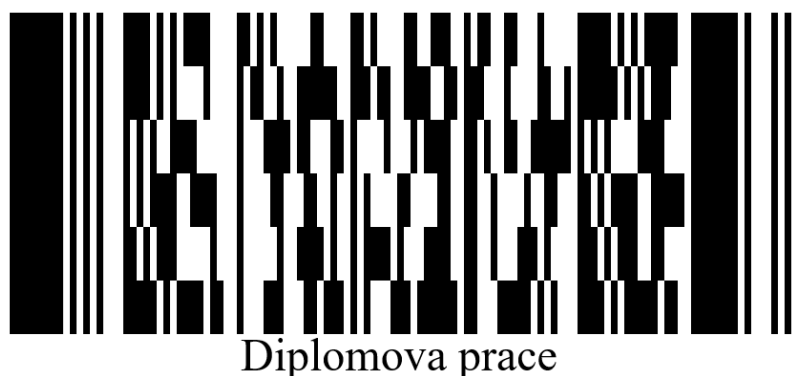
Obr. 6: Segmenty QR kódu [3]

PDF417

Kód s tímto názvem byl vyvinut v roce 1991. Je to jeden z prvních dvourozměrných kódů. Vzhledem se podobá jednorozměrným čárovým kódům, jen s přerušovanými čarami. Je to jediný z dvourozměrných kódů, který může být čten i lineárním nebo laserovým skenerem. Bylo tedy možné použít stávající čtečky s drobnými úpravami, které nemusely být nahrazeny kamerami. Kód se rychle rozšířil mimo jiné i díky tomu, že PDF417 má vysokou datovou kapacitu a schopnost detekce a opravy chyb. V roce 2010 se jednalo o nejpoužívanější dvoudimenzionální kód. [3]

Jedná se kód víceřádkový, ve kterém mohou být uložena data o různé délce. Jeho ukázkou můžeme vidět na Obr. 7. Data mohou obsahovat jak číselné hodnoty, tak text i ostatní binární data. Kód je tvořen počátečním a ukončovacím blokem, mezi nimi jsou data ohraničena pravým a levým indikátorem. Pravý a levý indikátor obsahují informace, jako je například číslo řádku, celkový počet řádků a úroveň schopnosti korekce chyb. [3]

Základní stavební jednotkou je datový blok, který začíná pruhem a končí mezerou. Tato datová jednotka je tvořena čtyřmi čarami a čtyřmi pruhy, jejichž šířka se rozděluje do 17 modulů. Právě od tohoto rozdělení je určeno jméno kódu. Počet řádků je variabilní v rozmezí 3 až 90. Každý z těchto řádků může obsahovat až 30 datových bloků. Poměr mezi řádky a datovými bloky může být různý. PDF417 tak může být rozměrově přizpůsoben dle dostupného tvaru prostoru. Poslední datové bloky v posledním řádku jsou vyhrazeny pro data potřebná ke korekci chyb. To dovoluje PDF417 aby zakódoval až 1108 bytů binárních dat, 1850 textových znaků nebo 2725 čísel. Tyto maximální hodnoty jsou ale zmenšeny o data potřebné ke korekci chyb. PDF417 může mít volitelný stupeň korekce chyb, a to v rozmezí 0 až 8. Minimálně obsahuje každý kód dva datové bloky použitelné pro korekci, maximálně pak 510 korekčních bloků. [5] [3]

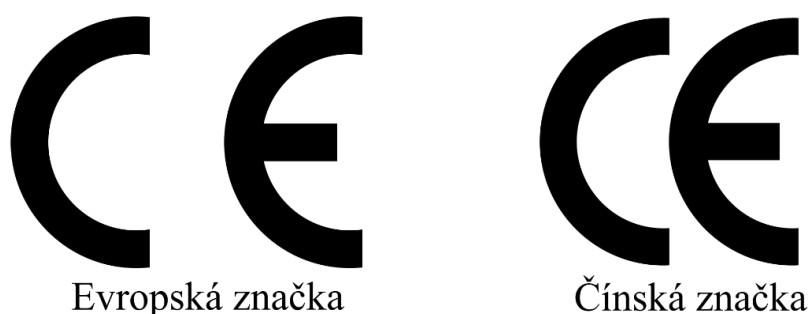


Obr. 7: Ukázka PDF417

2.4 Obrazová data

Dalším velice častým prvkem, který obsahují typové štítky, jsou obrazová data. Jednat se může o různá loga společností, výrobků či výrobních řad. Tyto loga většinou nemají žádnou důležitou funkci, jde tedy spíše o prestiž firmy.

Do obrazových dat se řadí i různé značky certifikací a další symboly. Příkladem může být značka CE, která se používá v EU pro výrobky, které splňují potřebné předpisy. Za zmínku stojí podobnost této značky se značkou China Export, která je nápadně podobná a liší se pouze vzdáleností znaků. Obě tyto značky jsou uvedeny na Obr. 8. [6]



Obr. 8: Ukázka CE značek [7]

Mezi další obrazová data lze zařadit různé výstrahy a také například zařazení výrobku do určité odpadové skupiny. Další velmi užitečnou informací mohou být instalační či manipulační pokyny ve formě piktogramů. Mimo tyto zjevné obrázky obsahují často typové štítky i další obrazová data v podobě různých rámečků, předělů, oddělovačů a podobně. Tyto data slouží pouze ke zpřehlednění typového štítku a zlepšení jeho vizuální podoby.

2.5 Požadavky průmyslu na vlastnosti a obsah štítků

Pro použití typových štítků v průmyslových aplikacích jsou kladeny zvýšené nároky na jejich životnost a čitelnost obsahu štítku na výrobku. Proto se mimo ceny štítku, také řeší vliv prostředí, se kterým je štítek v kontaktu, způsoby čtení štítku a mnoho dalších parametrů, které je nutné zohlednit.

Proto se například při použití dvoudimenzionálních kódů volí menší množství kódovaných informací, aby se předešlo zmenšování buněk, které by byly náchylnější na poškození. Také se často vyskytují na štítcích duplicitní data, kdy jednou jsou ve formátu pro strojové načítání a podruhé jsou v prostém textu, který umožňuje např. údržbě čtení typové informace lidským okem bez potřeby speciální čtečky. Obecně je v průmyslu rozšířenější použití Data Matrix kódu oproti QR, který se používá spíše na doplňkové a marketingové informace.

Velmi často lze například na zařízeních se síťovou kartou najít štítek s jejich MAC adresou jak v prostém textu, tak v nějakém typu kódu.

3 TISK TYPOVÝCH ŠTÍTKŮ

Pokud je stanoven obsah typových štítků přichází na řadu jeho fyzická realizace. V obecném případě se jedná o nějaký způsob tisku. Tiskové metody lze rozdělit dle několika kritérií. Jedním ze zajímavých kritérií je nosič typového štítku. Může se jednat o samotnou součást a v tom případě se tisk provádí přímo na ni. Nebo se může jednat o přidaný nosič v podobě papírového, plastového nebo kovového štítku. Mezi tiskové metody se řadí i obrábění, při kterém se odebere pouze materiál tvořící navrhnutý obraz typového štítku.

Většina typových štítků se omezuje pouze na jednobarevný tisk. Barevně se tisknou hlavně loga.

3.1 Média pro tisk štítků

První a nejjednodušší variantou je tisk přímo na součást, které náleží tisknutý štítek. Další variantou je tisk na určitý druh média, které se později připevní na součást. Mezi média, kterými se můžeme setkat velmi často, patří různé druhy samolepek. Dále se používají různé varianty kovových či plastových destiček, které se různým způsobem připevní na značenou součást.

Samolepky samotné se mohou lišit použitým materiálem. Kromě klasických papírových se používají i různé druhy odolnějšího papíru a také fólie. Povrchové úpravy výsledných samolepek mohou být také různé. Poté se liší jejich odolnost vůči vlivům prostředí a také cena. Ukázky samolepek jsou na Obr. 9.

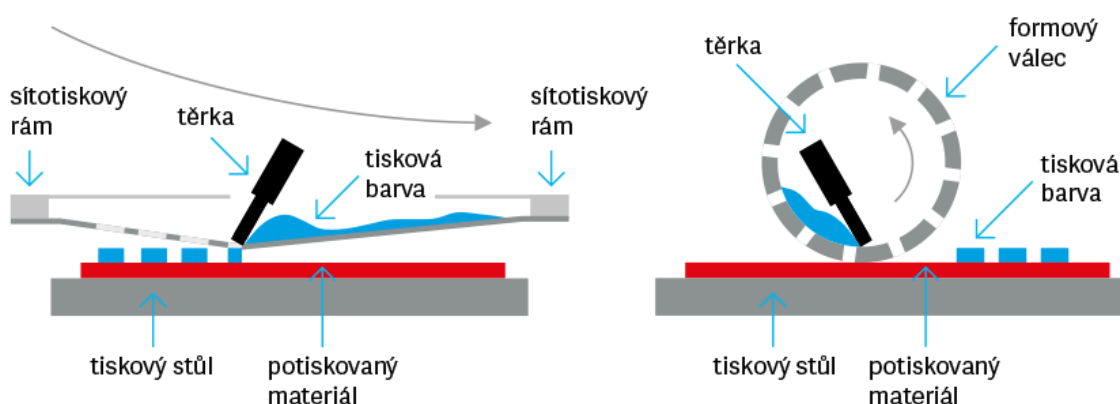
Někteří výrobci obcházejí technologickou neschopnost termotiskáren tisknout barevně tím, že při použití přímého termotisku napustí v požadované oblasti papír jinou látkou, která způsobí různé zbarvení při tisku. Nicméně je nutné si tyto podklady nechat vyrobit zakázkově. [8]



Obr. 9: Ukázka samolepek pro tisk [9]

3.2 Sítotisk

Mezi první často používané metody tisku se řadí metoda sítotisku. Při této metodě se využívá šablona z porézního materiálu, která je napnutá v rámu. Na síto se nanese barva, která se poté přetře třerkou. Během tohoto přetření se barva protlačí skrz šablonu na tiskový podklad. Tento princip je dobře viditelný na Obr. 10. Pokud je potřeba tisknout více barev je nutné zopakovat stejný postup pro každou barvu zvlášť ve vrstvách. Na každou barvu se používá vlastní šablona, proto vznikají problémy s přesným umístěním potiskovaného materiálu. Hlavním problémem této metody je možnost tisknout pouze statická data. Jde o šablonu, která je vždy vytvořená v jedné podobě a její úprava je časově náročná. Sítotisk je vhodný například na přípravu samolepek na dodatečný termotisk. Výhodou sítotisku je, schopnost tisknout na téměř jakýkoliv materiál. [10]



Obr. 10: Popis principu sítotisku [10]

3.3 Inkoustový tisk

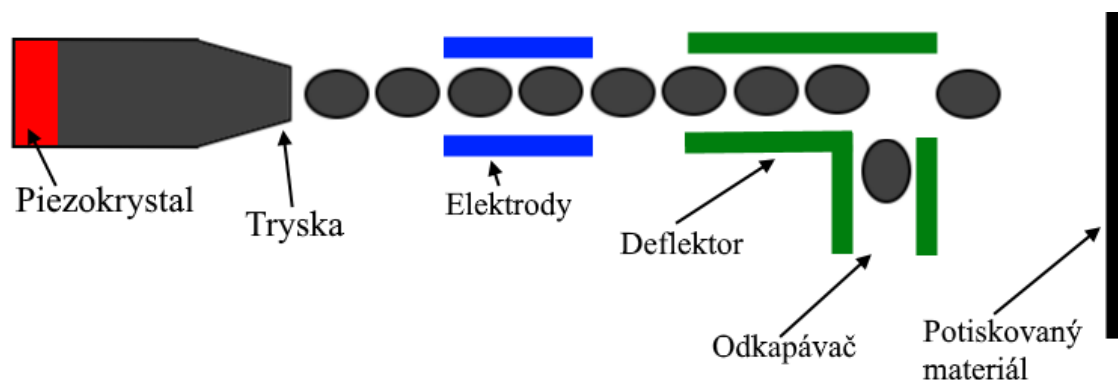
Při této technologii nedochází ke kontaktu tiskové hlavy s potiskovaným materiálem. Funguje na principu vytváření drobných kapiček, které se přenášejí na potiskovaný materiál v daném obrazci. Potiskovaný materiál může být libovolného typu. Častou nevýhodou je ucpávání trysky a také zasychání barvy v trysce. Další komplikací je, že pokud se tiskne velká plocha barvy na obyčejný papír, může dojít k jeho zvlnění během schnutí barvy.

Existuje několik principů tiskových hlav, během nichž se tvoří drobné kapičky. Tyto technologie se rozdělují do dvou hlavních skupin. Při první se tvoří souvislý proud drobných kapiček, ve druhé skupině se kapičky vytváří na základě pulzu. [11]

3.3.1 Souvislý proud kapek

Kapičky se vytváří z inkoustu procházející tryskou, která je rozvibrována pomocí piezokrystalu. Kapka získá elektrický náboj a poté její průchod elektromagnetickým polem určí, kam následně dopadne. Netisknuté kapky jsou odkloněny a vráceny do hlavního zásobníku.

Schematické znázornění celé hlavy je na Obr. 11. Tento způsob se využívá pro velmi rychlý tisk v průmyslu, většinou na průběžných linkách. Jsou jím například tisknuty data spotřeby na některé výrobky. Nevýhodou technologie je poměrně veliký rozměr kapek a tedy malé rozlišení tisku. [11]



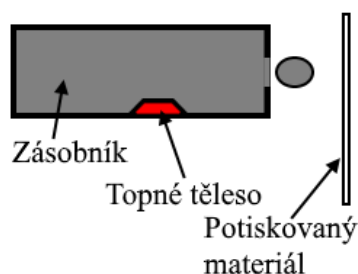
Obr. 11: Schéma hlavy pro tvoření souvislého proudu kapek

3.3.2 Kapky při pulzu

Jedná se o způsob tisku, při kterém se tvoří kapičky až dle potřeby na základě pulzu. Byl vyvinut v roce 1977 firmou Siemens. Dle způsobu tvorby kapiček lze rozdělit inkoustové tiskárny do skupin a to teplotní, piezoelektrické, elektrostatické a akustické. Vysoká přesnost a malý průměr kapiček je hlavní výhodou tohoto způsobu tisku.

Tepelná hlava

Při teplotní technologii tvorby kapiček se tvoří kapky během několika mikrosekund díky ohřátí topného odporu v zásobníku inkoustu. Díky rychlému ohřátí na relativně vysoké teploty (300-400 °C) se vytvoří bublinka, která vypudí kapku inkoustu z trysky. Schéma je na Obr. 12. Po vytvoření kapky se doplní nádržka s inkoustem, aby se proces mohl opakovat. Hlavy využívající tento způsob jsou levné, nicméně nevýhodou technologie jsou vyšší nároky na složení inkoustu.

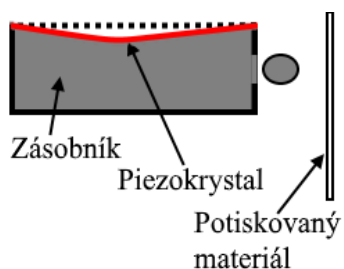


Obr. 12: Schéma hlavy založené na teplotním principu

Piezoelektrická hlava

Piezoelektrická technologie tvorby kapek se vyznačuje vysokou cenou za hlavu, ale má minimální požadavky na vlastnosti inkoustu. Kapka se vytvoří zkroucením piezokrystalu,

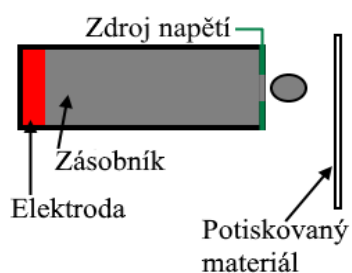
který vytlačí kapku skrze trysku. Jak je vidět na Obr. 13. Pomocí délky pulzu a velikosti trysky lze korigovat velikost kapky. [11]



Obr. 13: Hlava s piezokrystalem

Elektrostatická hlava

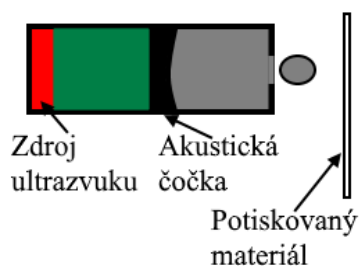
Elektrostatická technologie tvorby kapek vyřešila problém, kdy u piezoelektrické metody nešly vytvořit kapky menší, než byl otvor trysky. Díky elektrostatickým silám mezi hlavou a elektrodou v zásobníku se v inkoustu indukuje náboj. Takto nabitý inkoust je přitahován k hlavě, kde se poté oddělí ve formě kapky. Schéma je viditelné na Obr. 14. Metoda je schopná tisknout ve vyšším rozlišení než s piezoelektrickým uvolněním. Nicméně jde o technologii nerozšířenou. Má ji ve své nabídce jen několik firem. [11]



Obr. 14: Elektrostatická hlava

Akustická hlava

Akustická metoda tvorby kapek je moderní technika. Kapky jsou tvořeny pomocí akustické energie, která je formě ultrazvukového paprsku. Tento paprsek je soustředěn pomocí akustické čočky na povrch inkoustu, kde se vytvoří kapka. Trysky při použití této metody nejsou náchylné k ucpávání a také lze regulovat velikost kapky pomocí vzdálenosti čočky. Schéma i s čočkou je na Obr. 15. [11]



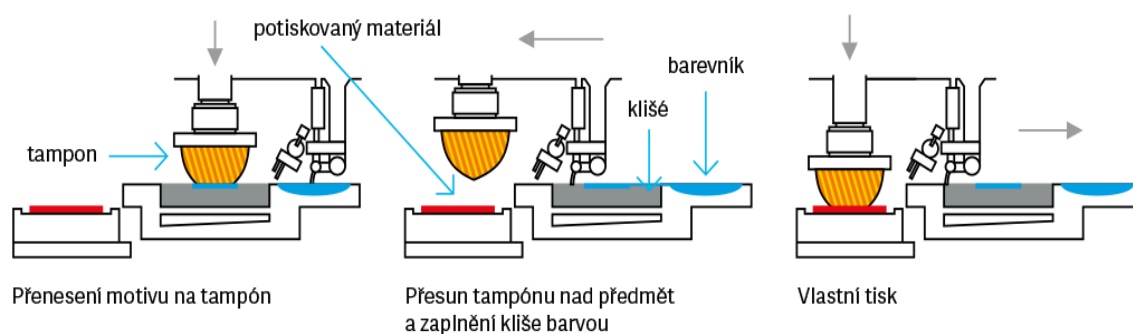
Obr. 15: Akustická hlava

3.4 Tamponový tisk

Tato metoda tisku je využitelná pouze na menší tisknuté obrazce. Předností je schopnost tisknout i na zvlněné případně lehce zakřivené povrchy, a to jak konvexně, tak konkávně. Tato technika využívá pružný tampon různých tvarů, který přenáší tisknutý motiv z klišé na potiskovou plochu. Klišé neboli vzor je pevně daný a je vytvořen pomocí chemického leptání nebo jinak mechanicky obrobena. Výhodou je, že na tisk se může používat poměrně tekavý inkoust, protože mokrá vzor je odhalený pouze malou chvílí.

Samotný princip tamponového tisku je znázorněn na Obr. 16. Tampon se namáčkne na navlhčené klišé a poté se přesune nad potiskový materiál. Během přitisknutí tamponu na potiskovaný materiál se zároveň vlhčí klišé. Při návratu tamponu nad klišé se navlhčené klišé setře. Poté se proces opakuje.

Podobně jako u sítotisku nelze současně tisknout několika barvami. Tato nevýhoda se řeší několika různými klišé vlhčenými různými barvami. Pokud je potřeba tisknout více barev je nutné tisknout je postupně. [12]



Obr. 16: Princip tamponového tisku [10]

3.5 Termotisk

Existují dvě základní metody termotisku, obě tyto teploty využívají zvýšené teploty k tisknutí. Prvním druhem je přímý termotisk, druhým je pak termotransferový tisk. Oba druhy této metody jsou široce zastoupeny při tisku typových štítků. Jedná se o štítky, které využívají nějaké potiskované médium. Potiskovaným materiálem při tisku typových štítků jsou nejčastěji různé formy samolepek. Kromě typových štítků se termotisk používá i na jinou velkou a často zastoupenou skupinu. Jedná se o tisk účtenek a dalších dokumentů.

3.5.1 Přímý termotisk

Jednodušší metodou termotisku je přímý termotisk. Při tomto způsobu tisku se pomocí tiskové hlavy lokálním zahřátím zaktivuje pigment obsažený v potiskovaném médiu. Účtenky a podobné dokumenty, k jejichž tisku se přímý termotisk využívá, jsou poměrně náchylné. Musí být uschovány mimo vyšší teploty, neboť by došlo k aktivaci pigmentu na jiných místech, než bylo určeno tiskárnou a tím k znemožnění čitelnosti dokumentu.

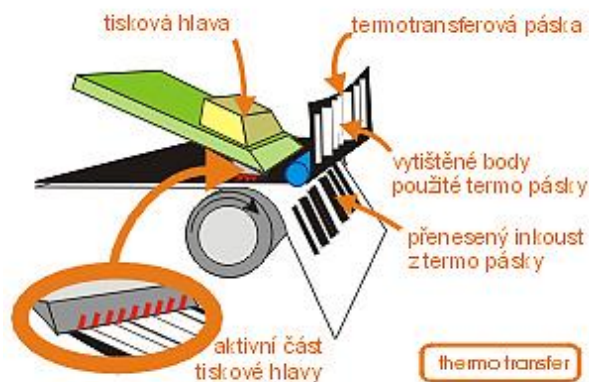
Mimo teplotu jsou náchylné i na UV záření, při jeho působení postupně blednou. Další nevýhodou je vyšší teplota, kterou musí vyvinout tisková hlava, kvůli aktivaci pigmentu a tím její vyšší opotřebování. Výhodou je, rychlost a to, že k tisku není potřebná nosná pigmentová páska. Schéma tiskové hlavy je viditelné na Obr. 17.



Obr. 17: Schéma přímého termotisku [13]

3.5.2 Termotransferový tisk

Druhou tiskovou metodou je nepřímý termotisk neboli termotransferový tisk. Při tomto způsobu se využívá fólie s pigmentem, při jejím zahřátí dojde k přenosu pigmentu mezi potiskovaným materiálem a nosnou páskou. Přenesený pigment se potom působením tlaku zalisuje do potiskovaného materiálu. Je potřebná další páska s pigmentem, ale takto vytištěné štítky či jiné dokumenty jsou mnohem stálejší.



Obr. 18: Schéma hlavy pro termtransferový tisk [13]

Oba způsoby jsou velmi často používány. Tiskárny ve většině případů umožňují oba způsoby termotisku a konkrétní technologii je možné zvolit v nastavení tiskárny. Tiskárny pro využití v průmyslu vyrábí například firmy Zebra, Datamax-O-Neil, SATO a CAB. Existují i ruční přenosné verze těchto tiskáren.

3.6 Laserové značení

Při této technologii není zapotřebí nanášet barvu na potiskovaný materiál. Nejedná se tak o tisk ve standardním smyslu. Jedná se o značení, při kterém dochází ke změně materiálu působením laserového paprsku. Materiály reagují na laserové značení různě. U některých dochází ke změně barvy a tím vytvoření viditelného vzoru, u některých dochází k odstranění části materiálu a tím se vytvoří viditelný vzor. Vzniká tedy omezení, na jaký materiál je tato technologie použitelná.

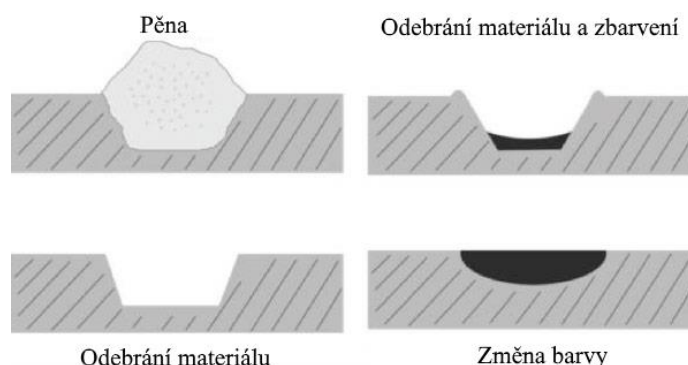
3.6.1 Značení plastů

Značení plastů laserem se liší oproti jejich potisku tím, že potisk je vždy pouze na povrchu plastového dílu. Zde je vystaven okolním vlivům, kterým může odolat plastový díl, ale inkoust použitý při tisku ne. Nejde jen o chemické působení okolí ale i o mechanické namáhání povrchu otěrem, kdy je potisk mnohem náchylnější než jednolitý plastový povrch. Laserové značení využívá několika dalších principů, které přispívají k čitelnosti i při vymytí či jinému odstranění inkoustu. Mohou také úplně nahrazovat inkoust tím, že poskytnou čitelnost vzoru za pomoci jiné kontrastní změny oproti okolnímu povrchu značené součásti. [14]

Při nízké intenzitě laseru, a správně zvoleném plastu dochází k vytváření bublinek v materiálu a tím k tvoření pěny, která udává tisknutý vzor. Tato pěna má světlejší barvu než okolí a vystupuje lehce nad povrch.

Další metodou je přímé odstraňování materiálu laserem. Dochází k lokálnímu ohřátí materiálu nad teplotu varu a k odpaření materiálu. Po zchladnutí zbývajících materiálu se vytvoří viditelný vzor podobně jako při leptání. Pro zlepšení čitelnosti lze upravit intenzitu laseru, aby došlo ke změně barvy plastu v oblasti odebíraného materiálu.

Při použití laserového záření o dostatečně krátké vlnové délce lze způsobit disociaci molekul plastu. Toto dělení molekul má za následek změnu barvy. Toto zbarvení lze zvýšit použitím vhodných přísad. Všechny technologie jsou znázorněny na Obr. 19. [14]



Obr. 19: Různé formy laserového značení [14]

3.6.2 Značení kovů

Pro kovy obecně existuje několik způsobů značení pomocí laseru. Při značení ocelí se využívá velmi krátkého lokálního zahřátí materiálu a vytvoření oxidu, který má tmavší barvu než okolí. V určitých případech lze ovlivnit zbarvení materiálu a tím vytvořit různé barvy. Při tomto způsobu značení se neodpařuje materiál a nedochází k jeho úbytku.

Pokud se využívá laseru pro značení eloxovaného hliníku, dokáže laser zanechat velmi kontrastní a viditelný vzor. Díky tomu, že lze pomocí kontroly výkonu laseru značit světlou i tmavou barvou, je možné využít laser na značení hliníku libovolné barvy. [15]

3.6.3 Značení elektronických součástek

Technologie značení laserem lze využít na potisk epoxidových těl elektronických součástek. Využívá efekt, že se epoxid při působení laseru dokáže lokálně natavit a vytvořit viditelně světlejší pěnu.

3.7 Ostatní mechanické značení

Používá se v případech, kdy je součást vystavena negativním vlivům prostředí v míře, která nedovoluje použití tisknutelných štítků a zároveň nelze značit laserem. Případně pokud nelze použít laser z jiných důvodů, lze vytvářet tisknutý obrazec jinými invazivními způsoby. Mimo ruční gravírování, které ze své podstaty a časové náročnosti nemůže být použito pro větší než kusovou výrobu, existuje také strojové mechanické gravírování. A tedy hlavně gravírování pomocí frézky. Dále je hojně využíváno i mikroúderové značení.

Mikroúderové značení využívá velmi tvrdý hrot, který je opakovaně zatlačován do materiálu v požadovaném tvaru. Takto vytvořený obraz je poměrně dobře čitelný. Tento způsob lze také využít k zaznamenání jiných než textových informací, a to zejména v podobě dvoudimenzionálního Data Matrix kódu.



Obr. 20: Ukázka Data Matrix kódu vytvořeného pomocí mikroúderů [16]

4 ZEBRA TECHNOLOGIES

Tato firma s celým názvem Zebra Technologies Corporation je výrobcem průmyslových termotiskáren, které používají tiskový jazyk ZPL II případně jeho starší verzi. Také tato firma dodává spotřební materiál pro tiskárny a další produkty. Mimo jiné vyvíjí softwarové řešení pro návrh štítků a další software.

4.1 Historie

Firma, která vytvořila jazyk ZPL a mimo jiné i celé portfolio termo tiskáren byla založena v roce 1969 jako Data Specialties Inc. Tato firma v začátku svého působení vyráběla přístroje pro vytváření a čtení děrných pásků. Po několika problémech a nezdarech vsadili zakladatelé na nově vznikající trh, který využíval čárové kódy pro značení výrobků.

Prvním velkým úspěchem bylo vytvoření tiskárny Zebra na tisk čárových kódů. Kvůli zhoršené odolnosti a potížím se čtením vytvořili svou první tiskárnu využívající termotransferový tisk. Díky této technologii byla zajištěna vyšší životnost tisknutých štítků a odolnost vůči vysokým či nízkým teplotám.

V roce 1986 došlo k přejmenování na Zebra Technologies Corporation. Poté byla výroba přesunuta zpět na území USA z Japonska, kam byla umístěna z důvodu nižších výrobních nákladů. Následovalo rozšíření společnosti i do Evropy, mimo jiné i do Brna.

V dnešní době je Zebra dodavatelem jak tiskových tak dalších řešení do průmyslu, ale i do zdravotnictví a dalších odvětví. Ukázka moderní tiskárny ZT400 je viditelná na Obr. 21, modely jí podobné lze najít v Brně v několika stovkách výskytů. [17]



Obr. 21: Ukázka moderní tiskárny Zebra ZT400 [18]

4.2 ZPL II

ZPL neboli Zebra Programming Language je jazyk vyvinutý firmou Zebra Technologies pro jednoduchý a rychlý návrh a následný tisk typových štítků. Postupem času, došlo k vytvoření druhé generace tohoto jazyka neboli ZPL II. Obě varianty spolu nejsou plně kompatibilní. Z důvodu zvýšení rychlosti tisku bylo nutné pozměnit způsob vytváření i interpretace výsledného skriptu.

Existuje několik způsobů vytváření skriptů. Zebra poskytuje nástroj pro vytváření štítků pod názvem ZebraDesigner, který je podrobněji rozebrán v následující kapitole. S poměrně malou znalostí jazyka a s dostupným manuálem lze skripty velmi snadno vytvářet i v jednoduchých textových editorech.

Pro nenáročnost se skript skládá z jednotlivých příkazů, které jsou vždy uvedeny klíčovým znakem. Klíčový znak se liší pro obecné tiskové příkazy a pro složitější příkazy, které většinou spouští určitou funkci. Klíčový znak pro funkce je „~“ pro ostatní tiskové operace je to „^“. Po tomto úvodním znaku musí následovat většinou dvouznakový identifikátor požadované funkce či vlastnosti s parametry.

Existuje i několik dalších firem, které převzaly funkcionalitu jazyka ZPL II, a vytvořily pro vlastní tiskárny překladové nástroje. Nicméně jejich podpora nemusí být dokonalá, řadu příkazů může úplně vypouštět a podobně. Takovou firmou je například CAB. [19]

4.2.1 Základní struktura

Je zvykem každý jednotlivý štítek uzavírat do dvojice znaků ^XA a ^XZ. Tyto znaky ohraničují jednotlivé zprávy. Vše, co je uzavřeno mezi těmito znaky, je tisknuto najednou. Nicméně v praxi je někdy možné vidět jeden štítek složený z více takto uzavřených zpráv. Poté se používá jedna hlavní zpráva, která obsluhuje samotný tisk a předchozí nebo následující zpráva přepisuje interní nastavení nebo tiskové nastavení tiskárny.

Tisknuté části štítku se uvozují příkazem pro nastavení umístění na štítku. Jako jednotka pro měření vzdálenosti se nejčastěji používá tisknutelný bod, který má různou velikost dle použité tiskové hlavy. Možnost určení pozice je dvojí. První možností je před tisknutou část umístit příkaz ^FO, který určuje pozici následující části. Druhou možností je definovat tyto části v jiném souboru, poté načíst jeho rozložení a pomocí příkazu ^FN určit číslo části a jen vyplnit její data. [20]

4.2.2 Tisknutelný obsah

ZPL II je připraven pro tisk téměř všech typů obsahu typových štítků. Lze snadno tisknout texty, čáry, grafiku, dvoudimenzionální i čárové kódy a další.

Tisk textů

Pro tisk textů je základní výběr fontů. Tyto fonty lze vybrat pomocí ^A následující znak (A-Z, 1-9) tohoto příkazu je vybraný font, který se bude používat k dalšímu tisku. Také lze zvolit výchozí font, který bude použitý při tisku, kdy není definován jiný font. Velké

nebezpečí při použití fontů je, že jednotlivým znakům pro výběr lze přiřadit vlastní font, přičemž toto nastavení se zapamatuje i pro následující štítky. [20]

Tisk čárových a dvoudimenzionálních kódů

ZPL II má širokou podporu pro tisk čárových i dvoudimenzionálních kódů. Zvolí se typ a velikost kódu, případně další vlastnosti a poté se zadají data, která mají být uložena v kódu. Překódování se provádí až před tiskem přímo v tiskárně. Z tohoto důvodu je nemožné použít vlastní druh kódu. [20]

Tisk grafických dat

Jazyk ZPL II má definované jednoduché nástroje pro vykreslování tvarů jako jsou čáry, kruhy, obdélníky a podobně. Mimo takto předem definované tvary lze tisknout i jinou grafiku. Tato grafika, se tiskne ve formě bitmapy a dovoluje i komprimovaný zápis. Bohužel tato komprimace není popsána v žádné dostupné literatuře. A jediná možnost je použít dodávaný editor štítků pro využití této komprimace. [20]

4.3 ZebraDesigner

Takto se jmenuje originální software vyvinutý pro návrh a tisk štítků pomocí tiskáren Zebra. Existuje ve dvou variantách a několika verzích. Zde popsána verze je 2.5.0 (Build 9393). První variantou je demoverze pro vyzkoušení, která je dostupná na stránkách Zebra Technologies po registraci. Omezením demoverze je, že každý tisknutý znak „e“ je nahrazen znakem „Z“. A všechny grafické prvky obsahují ve výsledném tisku vodoznak firmy Zebra. Demoverze je určena pro vyzkoušení funkcionality softwaru, ne pro jeho průmyslové nasazení. Druhou variantou je Pro, která nabízí rozšířené možnosti a je zpoplatněna. Smyslem softwaru je nabídnout jednoduché uživatelské rozhraní pro návrh štítků a jejich následný tisk. [21]

Samozřejmě, že existují i jiné editory pro návrh štítků. Nicméně tyto editory nejsou oficiálním řešením firmy Zebra Technologies. Mezi tyto programy patří například CodeSoft, který je zpoplatněn ve stovkách eur za licenci. [22]

4.3.1 Výhody softwaru

Software je stavěn robustně a je vyvíjen výrobcem tiskáren, což pro koncového uživatele znamená, že kompletní portfolio tiskáren bude s tímto softwarem fungovat. Software dokáže upravit nabízené možnosti pro právě zvolenou tiskárnu a prvky přizpůsobit konkrétnímu modelu a jeho schopnostem. Pro data, která jsou variabilní, nabízí několik způsobů, jak je nahradit aktuálními daty.

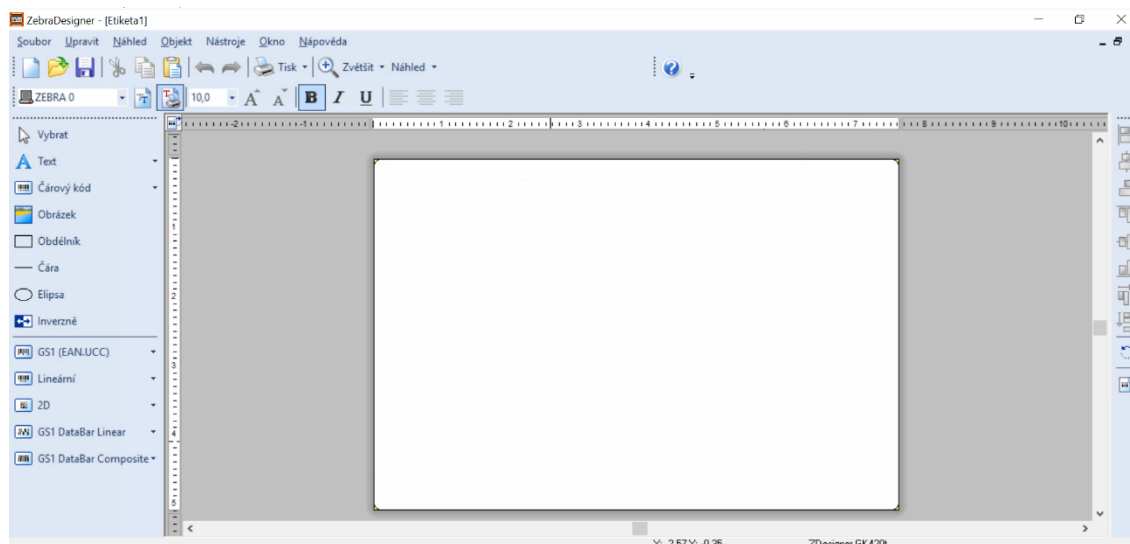
4.3.2 Nevýhody softwaru

Velkou nevýhodou je nutnost používat tento software pro kompletní tisk. Vnitřně se sice štítky překládají do ZPL nicméně tento výstup téměř nelze získat. A proto je téměř nemožné tento software používat jako zásuvný modul v jakékoliv aplikaci. Vytvořené štítky mají vlastní formát souboru. Přenášení navrhnutých štítků je tedy téměř nemožné.

Další nespornou nevýhodou je nutnost platit za tento software, neboť demoverze je kvůli svým omezením nepoužitelná.

4.3.3 Práce s programem

První věcí po instalaci a spuštění programu je výběr typu tiskárny, na který se štítky navrhuji. Po zvolení a případné instalaci ovladačů pro tiskárnu je uživateli zobrazeno základní okno programu, které slouží pro návrh. Toto okno je viditelné na Obr. 22.



Obr. 22: Základní pracovní okno programu

Pod titulkem okna, kde je název programu a také název aktuálního štítku se nachází standardní nabídka Soubor, Upravit a podobně. Pod touto nabídkou je panel nástrojů pro práci se štítkovým souborem a také pro tisk. Ještě níže se nachází panel pro nastavení textu. V levé části je Panel pro výběr vkládaného obsahu. Oddělovačem se rozdělují čárové a dvoudimenzionální kódy od ostatního obsahu. Uprostřed je samotný náhled štítku a na pravé straně je nástroj pro nastavení pozice a práci s jednotlivými objekty.

Každý vkládaný obsah se nastavuje pomocí průchozího dialogu, který umožňuje nastavit požadované parametry vkládaného objektu. Po vložení se vykreslí na náhledové ploše, kde je možné měnit mu pozici a také velikost, pokud je to možné.

Další nastavení tiskárny je dohledatelné přes vrchní menu. Toto nastavení umožňuje měnit základní odsazení štítku v tiskárně, jeho velikost a podobně. Také tento program umožňuje posílat příkazy do tiskárny pro vytisknutí konfiguračních parametrů případně pro tisk obsahu různých složek, fontů a podobně.

Na panelu ke konfiguraci textu je patrné, že lze zvolit i fonty, které jsou na aktuálním počítači. Pokud budou použity při tvorbě štítku, je nutné použít další softwarový nástroj pro jejich nahrání do tiskárny. Mohou ale vznikat problémy s licenci fontů, kdy mohou být použitelné pro tisk, ale ne pro nahrávání a tedy další šíření.

5 NÁVRH EDITORU TYPOVÝCH ŠTÍTKŮ

Navrhovaný software je určen pro integraci do stávajících řešení obsluhující výrobní linky. Pověřený uživatel by měl mít možnost změnit či vytvořit výrobní štítek, pro který má často pouze výkresovou dokumentaci. Takto vytvořený štítek, lze pak snadno využívat v procesu výroby. Při stávajícím řešení je nutná znalost jazyka ZPL II na vysoké úrovni, pokud je potřebná jakákoliv změna na štítku. Také není zavedena jednotná koncepce návrhu štítků a tím dochází k problémům při jejich úpravách. Dalším důvodem je snadné načítání dat z databázových systémů. Oproti řešení třetích stran obsahuje editor pouze funkce, které jsou potřebné pro stávající implementaci nadřazeného systému. Díky této vlastnosti je cena editoru nižší než u editorů třetích stran.

Jak bylo již zmíněno v kapitole o jazyku ZPL II nastavení štítků zůstávají v tiskárně i pro další štítky. Tato vlastnost nepředstavuje problém, pokud je tiskárna určena k sériovému tisku jednoho typu štítků bez dalších požadavků. Problémem se tato vlastnost stává až v případě, že dojde k situaci, kdy je nutné využít tiskárnu pro tisk více typů štítků. Je tedy nutné preventivně vše přenastavovat a vzhledem k tomu, že některé štítky mohou být vytvořené přes ZebraDesigner není jasné, jaká nastavení jsou posílána do tiskárny. Tato neznalost poté vede k zbytečnému narůstání obsahu štítků, neboť je nutné podchytit všechny eventuality, které mohly změnit nastavení kritické pro následující štítek. Některá nastavení lze z tiskárny vyčíst, bohužel ne všechny.

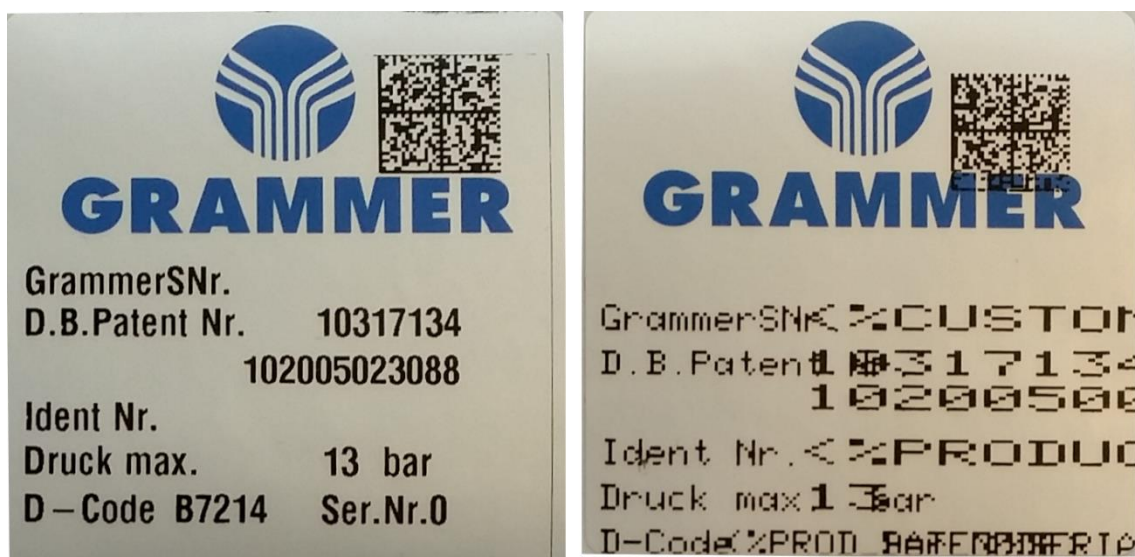
Dalším problémem se stávají fonty. Neschopnost některých tiskáren Zebra Technologies vyexportovat jednou nahrané fonty představuje velký problém hlavně v případě poruchy tiskárny. Tiskárna se snadno nahradí za jiný kus, nicméně bez znalosti, jaký font byl uložen v paměti a jak byl nalinkován, se z jednoduché výměny stává operace, která může zabrat i několik hodin.

5.1 Koncepce editoru typových štítků

Editor může využívat jeden ze dvou základních přístupů ke generování štítků a volba přístupu ovlivňuje koncepci editoru.

První přístup je použití plné formy jazyka ZPL II, kdy vygenerovaný štítek obsahuje příkazy s parametry. Renderování grafiky štítku provádí tiskárna na základě právě těchto příkazů a parametrů. Tento přístup vyžaduje, aby tiskárna byla správně nastavena a obsahovala potřebné fonty a k nim příslušné nastavení. Fonty často nejsou v tiskárně pojmenovány tak aby byly dohledatelné. Tím vznikají problémy při jejich nahrazení a dalším využívání. V případě poruchy tiskárny nelze písma určit a musí být nahrazeny jiným písmem, který lze vybrat pouze na základě grafické podobnosti. Také nelze snadno kontrolovat, jestli jsou splněny všechny licenční podmínky používání fontu, pokud není známo jeho jméno. Případně při využívání návrhu rozložení obsahu v odděleném souboru musí být tato kostra přítomná v tiskárně.

Při použití prvního způsobu jsou kladeny menší nároky na přenos dat, neboť se posílají pouze data, jaký obsah se má ve výsledku vykreslit. Problémy s nastavením a fonty lze pozorovat na Obr. 23.



Obr. 23: Ukázka správného (vlevo) a špatného (vpravo) tisku

Druhý přístup je provést renderování grafiky v překladači při generování štítku a do tiskárny posílat bitmapu s počátkem štítku. To výrazně zjednodušuje konfiguraci tiskárny. Při tomto způsobu je použití jakéhokoliv písma a rozložení štítku bez problémů. Mohou tak ale narůstat data, která musí být přenesena do tiskárny pro potřeby tisku. Tuto nevýhodu lze potlačit použitím komprimačních algoritmů. Porovnání jednotlivých zápisů je viditelné na Tab. 1.

[illegible]

Tab. 1 Jednotivé formy zápisu typového štítu

Z důvodu maximální kontroly nad tisknutým obsahem bylo rozhodnuto o využití druhé možnosti a to vygenerovat celý štítek těsně před tiskem ve formě obrázku. Po tomto kroku následuje do ZPL II a tisk.

Bylo tedy nutné navrhnout jak software pro překlad již vytvořeného štítku, tak software pro vytváření štítků a jejich ukládání. Samotný štítek je pak ukládán do souboru ve formátu xml, který je přenosný a tím pádem je možné štítky vytvořit na jiném místě, než kde budou ve finále použity a tisknuty.

Vzhledem k tomu, že ve výrobě se při tisku využívají variabilní data, bylo nutné ve štítku vytvořit určité značky, které se až těsně před překladem do ZPL II nahrazují konkrétními daty. S tím souvisí také nevyužití vestavěných generátorů pro čárové ale i dvoudimenzionální kódy. Jejich využití by bylo možné nicméně by to nabouralo koncepci maximální kontroly nad generovaným obsahem.

5.2 Základní rozbor ergonomie

Jakýkoliv editor je nutné navrhovat s ohledy na úroveň a požadavky uživatele. Všechny editory, ať už se jedná o textové, grafické či další zaměření, pracují s obdobným rozložením ovládacích prvků. Vždy je přítomen nějaký druh náhledu editovaných dat. Další část pak tvoří samotné editovací nástroje. Přidanou částí může být určitý druh seznamu obsahu nebo například pro obrazové editory seznam obrazových vrstev. Pro jednotlivé editovací nástroje mohou existovat přesnější nastavovací okna, pokud je tyto nástroje vyžadují. V těchto samostatných oknech už často není struktura a rozložení ovládacích prvků standardně daná. Zde se nejvíce rozlišuje mezi editory pro amatéry a profesionály.

U profesionálních verzí editorů je možné detailní nastavení editovací funkce. Nicméně takováto nastavovací okna se mohou stát pouze obrovskou tabulkou pro vyplnění dat, které ale z uvedených popisků nelze jednoznačně, ale hlavně jednoduše určit. Tato počáteční nevýhoda je nicméně vynahrazena širokou paletou způsobů použití dané funkce. Příkladem profesionálního editoru může být například program pro návrh vestavných skříní, který je na Obr. 24.

Parametre

vstavanej skrine

- NONAME.IBX

Definícia skrine

Popis

Šírka

Výška

Hĺbka

3000

2100

600

Rovnaká šírka

☐ dielov

☒ dverí

☐ oboch

Dosky, úkopy

Rampa

Zákl.dosky

☒ Vrchná
☐ Spodná
☒ Ľavá
☒ Pravá
☐ Zadná

Priečky

☒ A/N

Výška

Hĺbka

0

0

X-posun

Y-poz

Z-poz

0

0

0

☐ Stípiť

Počet

Aktívny diel/dvere

dielov

dverí

3

3

☒ 1
☐ 2
☐ 3
☐ 4
☐ 5
☐ 6
☐ 7

Dvere

Filter dverí

Materiál dverí (výplne)

ALL

lamino - borovica sukatá príř.

Filter rámov

Farba a typ rámov

ALL

bezný (lamino)

Výška

Šírka

X-posun

Z-pozícia

Delenie

0

0

0

0

0

Prekryv

Typ

Vôra1

Vôra2

30

1

6

60

☐ 4 kolaje

Typ dverí

☒ posuvné
☐ skladací
☐ otočné
☐ obyčajné
☐ žiadne

Police

Počet

X-poz

Šírka

Hĺbka

Y-poz

Hrúbka

0

0

0

0

0

Z-pozície

Uchyt. (ukpetovr)

Tvar (ozrt)

Prevedenie (123)

Predvŕtanie (01)

Zdvojenosť (12)

Parametre

Zásuvky

Počet

Z-pozície

☐ Kovové

0

X-poz

Y-poz

Šírka

Hĺbka

0

0

0

Výšky

Zdvojenosť (12)

Ťače, výsuvné a sklopné vešiaky

Počet

Šírky/hĺbky

Priemer

Typy (tvskK)

0

30

Z-pozície

Y-pozície

X-pozície

Ďalšie prvky

Príslušenstvo

Materiál (hranenie)

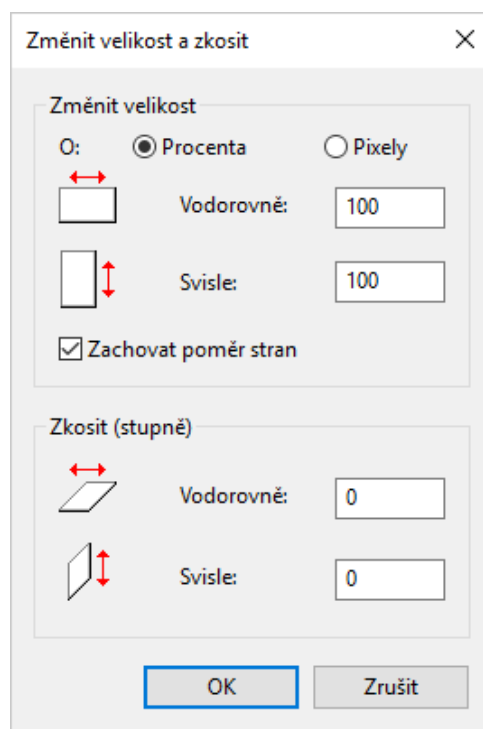
Ceny

Cancel

OK

Obr. 24: Nastavovací dialog program INBOX [23]

Na rozdíl od profesionálních verzí je u amatérských programů nastavení často přesně popsáno i s obrazovými ukázkami. Nicméně tyto funkce budou jednodušší a nemusí mít přesně požadovanou funkcionalitu. Případně se výsledného efektu musí dosáhnout použitím více funkcí tam, kde u profesionální verze stačila pouze jedna funkce. Příkladem takto jednoduššího editoru může být program Malování, který je obsažen v operačním systému Windows. Jeho jednoduchý dialog s piktogramy na popis funkcí je uveden na Obr. 25.



Obr. 25: Ukázka jednoduchého dialogu.

6 REALIZACE EDITORU TYPOVÝCH ŠTÍTKŮ

Navrhovaný editor není určen pro úplné amatéry. Očekává se znalost problematiky tisku typových štítků, možnosti a provedení návrhu štítku v editoru. Také je potřebná znalost schopnosti generování jak dvoudimenzionálních, tak čárových kódů. Aplikaci lze rozdělit do tří částí.

V první části jsou jednotlivé prvky, které mohou být použity na štítku. V této části jsou obsaženy funkce pro kompletní využití požadovaného prvku. Tyto prvky budou rozepsány v další podkapitole.

Druhou částí je samotný editor, který umožňuje například tvorbu samotných typových štítků, jejich náhled, načítání a jiné. Využívá funkce použitých prvků z první části.

Poslední část se využívá před samotným tiskem. Dochází k předzpracování, což znamená náhradu dat, a překlad do jazyka ZPL II.

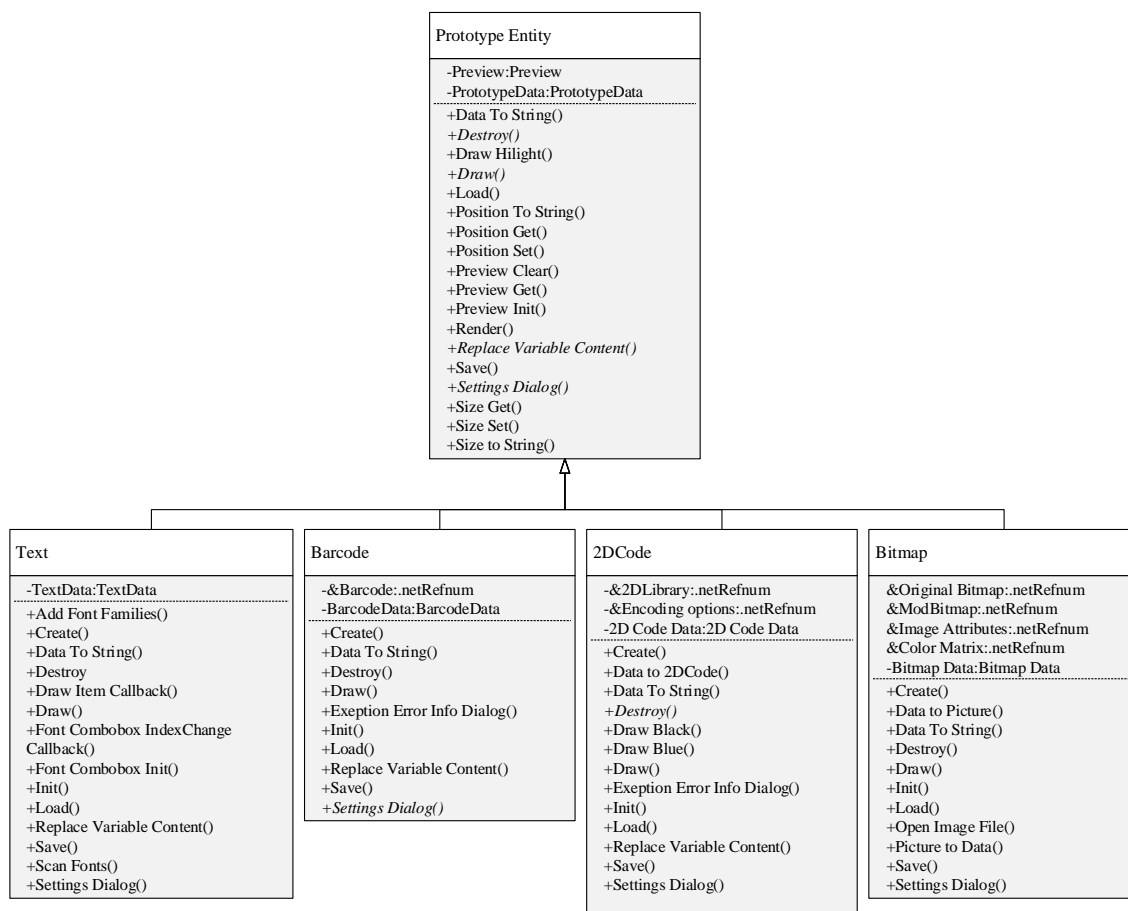
Celý software je naprogramován v LabVIEW, to je podmíněno tím, že aplikace, do kterých by měl být editor zastavěn, jsou ve vytvořeny ve stejném vývojovém prostředí. Od počátku návrhu bylo rozhodnuto, že aplikace bude provozována pouze na PC platformě, a tedy je možné využít funkcionalitu, kterou nabízí .NET Framework. K tomuto rozhodnutí bylo přistoupeno z důvodu, že funkcionalita, kterou nabízí LabVIEW týkající se dvoudimenzionální grafiky je nedostatečná. Byla by možnost použít placené rozšíření Vision, nicméně bez dalšího využití pro zpracování obrazu by tato platba byla zbytečná.

Použití .NET Frameworku je zdarma a je výchozí součástí moderních systémů Windows. Dalším důvodem bylo využití knihoven pro generování čárových a dvoudimenzionálních kódů. Tyto knihovny jsou ve formě dll a jsou přístupné právě přes rozhraní pro .NET. Vzhledem k tomu, že knihovny vrací obraz ve formě .net Image je toto využití výhodnější. Bylo by možné získaný obraz překreslit do standartního obrazového formátu, který využívá LabVIEW, nicméně tato operace by byla náročná a nepřinesla by žádné výhody.

Nevýhodou používání .NET Frameworku v prostředí LabVIEW je nutnost starat se o paměť. Bylo by sice možné použití automatického čištění paměti, nicméně to je možné pouze při zavření aplikace, a zbývající data mohou způsobovat problémy. Bohužel bez garbage collectoru, který je standardně využíván při použití funkcí .NET Frameworku v programovacím jazyce C# je nutné hlídat všechny alokace a poté uzavírat reference. Je tedy nutné i v případě využití triviálních funkcí, například pro použití určité barvy vložit blok uzavření reference na vhodné místo, aby byla uvolněna paměť.

6.1 Obsahová část

První částí celého softwarového systému je ta, která se stará o správu obsahu. Není předpokládána samostatná funkčnost této části a prvky z ní slouží pro použití v části editor a překladač. Obsahová část svými jednotlivými prvky odpovídá kapitole 2. Obsahuje tedy několik tříd, které spravují textové údaje, dvoudimenzionální i čárové kódy, a také obrazová data. Mimo tyto třídy obsahuje ještě jednu třídu nadřazenou, ze které jsou všechny ostatní podděny. Tato třída, která se nazývá Prototype, obsahuje data a funkce, které jsou společné pro všechny obsah. Na diagramu Obr. 26 je možné vidět celou strukturu obsahové části včetně metod a dat jednotlivých tříd. Metody jsou pro přehlednost uvedeny bez návratových hodnot a parametrů.

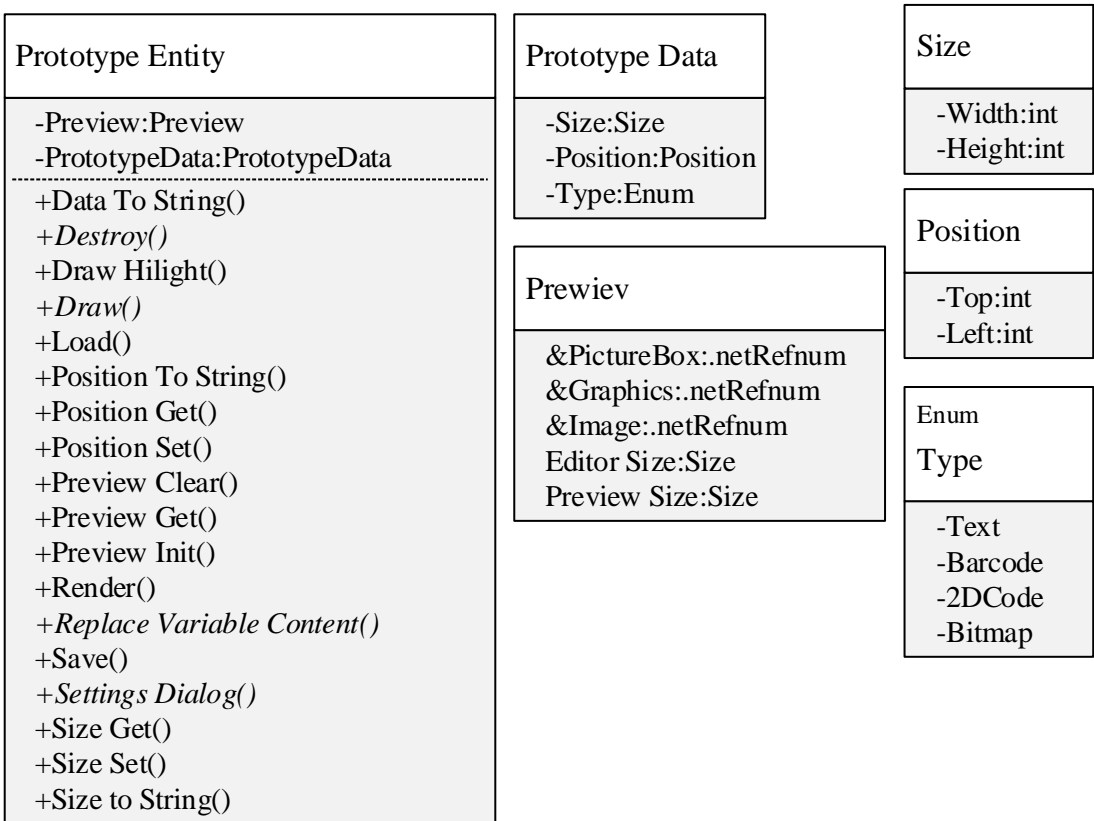


Obr. 26::Schéma obsahové části

6.1.1 Prototype

Třída byla navržena jako základ. Jedná se o třídu, která má vlastnosti třídy abstraktním. Obsahuje tedy několik metod virtuálních, které konkrétně implementují třídy podděně. Také obsahuje základní data, které se využívají pro všechny podděně třídy. Obsluhuje také inicializaci a čištění náhledů. Výčet metod a také obsažených dat včetně typů je na diagramu Obr. 27.

Díky schopnosti LabVIEW volat na vyžádání metody předků jsou takto volány metody pro ukládání a načítání ze souboru. Díky této vlastnosti se uloží jak data Prototype tak data instance třídy, která tuto metodu volala.



Obr. 27: Metody a data třídy Prototype

Mimo metody, které slouží k převodu dat do textové podoby, což je využito pro snadný zápis do seznamu, obsahuje třída Prototype několik metod pro čtení a zápis dat (metody Get, Set). Tyto metody jsou potřebné pro další použití těchto dat v podděněných třídách. Data jsou označena příznakem private, a tedy není jiný způsob, jak se k nim dostat přímo.

Pro nastavení parametrů děděných tříd je v nich konkrétně implementována metoda Settings Dialog, která vyvolá dialog pro nastavení. Zobrazené nastavení odpovídá potřebám jednotlivých implementací.

6.1.2 Text

Mezi nejčastěji využívaný obsah typových štítků patří textové údaje. Tato třída proto obsahuje data a metody pro práci právě s textovými údaji. Jejich rozvržení je viditelné na Obr. 28.

Textu je možné nastavit na libovolnou velikost a font, dále je možné nastavit příznak variabilních dat a jejich tag pro budoucí nahrazení. Přenositelnost textu je omezena pouze použitým písmem. Je nutné, aby použitý font bylo přítomný i na počítači, na kterém bude probíhat další zpracování štítku.

Text	TextData	Orientation
<div>-TextData:TextData</div> <div> +Add Font Families() +Create() +Data To String() +Destroy +Draw Item Callback() +Draw() +Font Combobox IndexChange Callback() +Font Combobox Init() +Init() +Load() +Replace Variable Content() +Save() +Scan Fonts() +Settings Dialog() </div>	<div> +Font Size:int +Font:string +Text:string +Variable Content: boolean +Tag:string +Orientation:Enum </div>	<div> Horizontal Vertical </div>

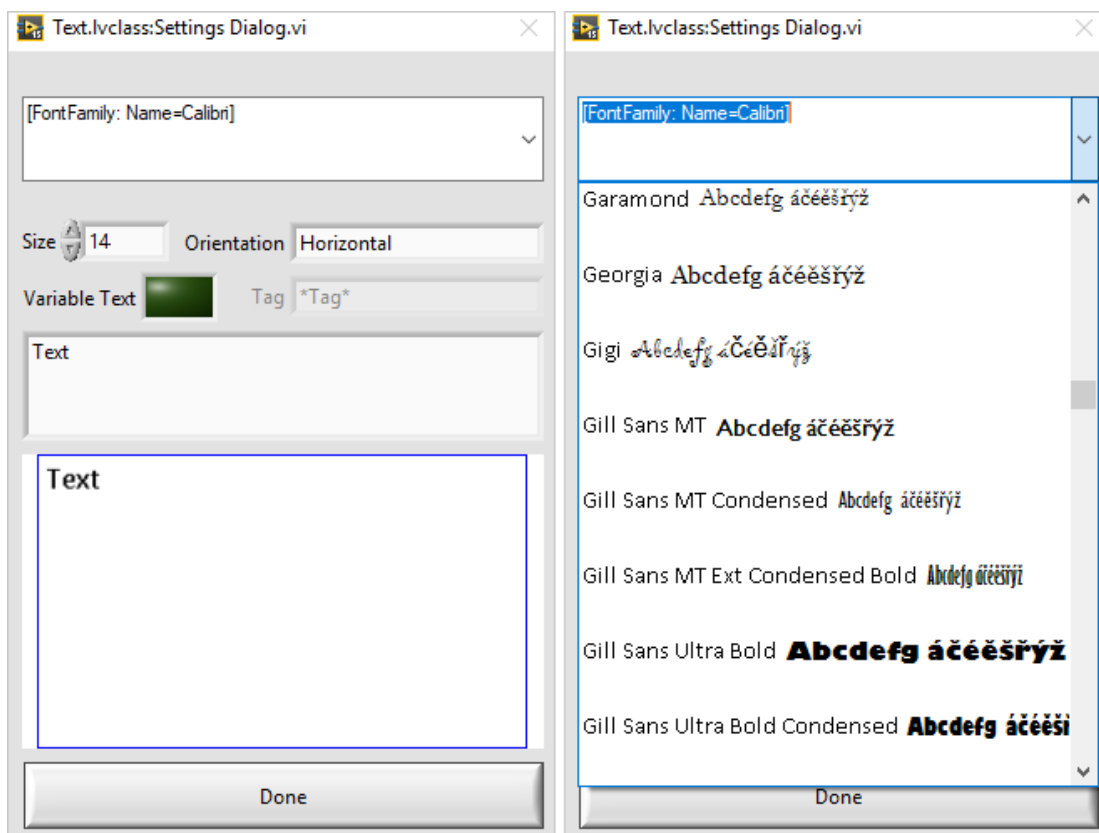
Obr. 28: Metody a data třídy Text

Pro snadný výběr z nainstalovaných fontů byl vytvořen speciální control, který poskytuje ukázkou fontu. Bylo ovšem nutné tuto funkcionalitu naprogramovat, k tomu se využívá schopnost LabVIEW převzít obsluhu událostí. Díky této schopnosti bylo možné při volání události na vykreslení položky ve výběru převzít její obsluhu a vykreslit mimo název písma také jeho ukázkou. Nicméně tato funkcionalita není dokonalá. Například vybraný font není v prvním řádku seznamu vykreslen správně a využívá ke svému zobrazení patrně jinou funkci, kterou nebylo možné dohledat. Taky lze spatřit nevýhodu v absenci vyhledávání písma pomocí psaní jeho jména.

Díky špatné detekci události při změně vybraného písma bylo nutno převzetí události, a to právě při změně voleného písma.

Pokud je zvolena volba variabilního textu zpřístupní se nastavení tagu, který slouží pro náhradu dat. Tato volba je také indikována modrým zbarvením ukázkového textu, aby byl snadno rozlišitelný.

Na Obr. 29 je viditelný dialog nastavení i ukázka fontů. V jeho spodní části je tlačítko pro potvrzení nastavení. Nad ním se nachází ukázka textu, kde modrý rámeček je celková velikost štítku, aby bylo možné nastavit text do požadované velikosti.



Obr. 29: Dialog nastavení textu v levé části i s ukázkou fontů v pravé části

6.1.3 Čárové kódy

Kvůli využití čárových kódů bylo nutné využít knihovnu pro jejich generování. Byla zvolena volně dostupná knihovna s názvem Barcode. Její autor ji zpřístupnil bez poplatku a pod volnou licencí Apache Licence 2.0. Na Obr. 30 je viditelné rozložení metod a dat, které se využívají pro návrh a vykreslení čárových kódů.

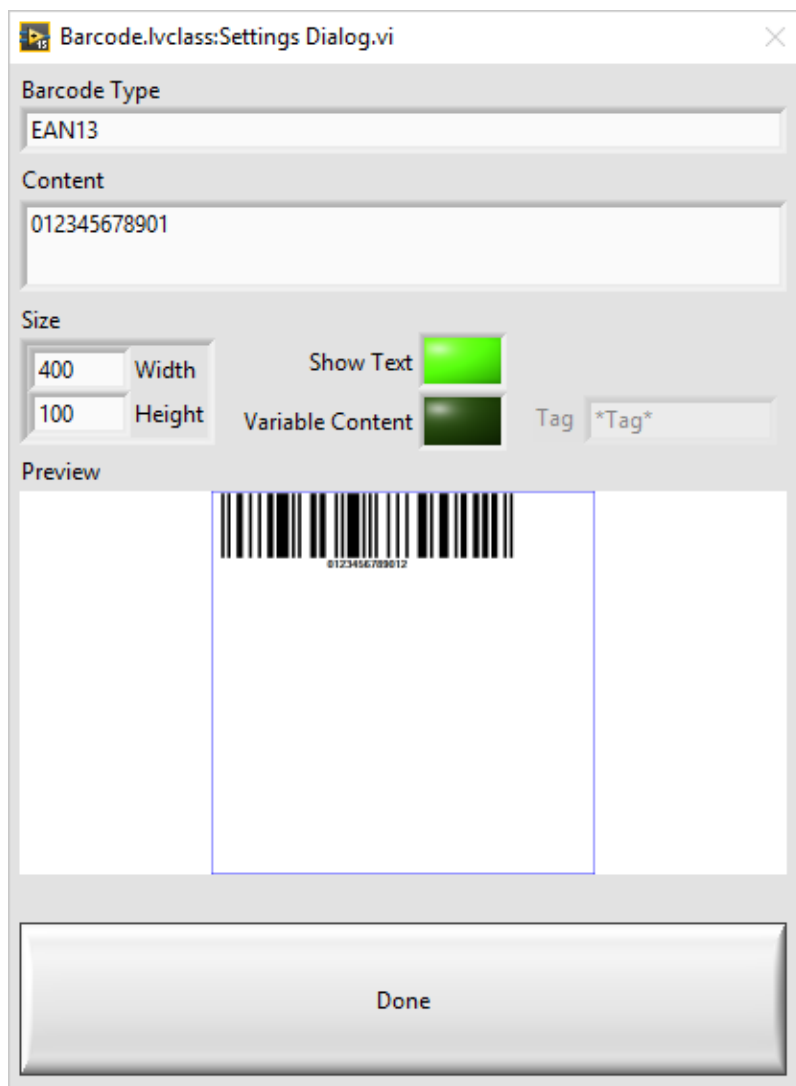
Barcode	BarcodeData	Barcode Type
-&Barcode:.netRefnum -BarcodeData:BarcodeData <hr/> +Create() +Data To String() +Destroy() +Draw() +Exeption Error Info Dialog() +Init() +Load() +Replace Variable Content() +Save() +Settings Dialog()	+Barcode Type:Enum +Content:string +Show Text :boolean +Variable content:boolean +Tag:string	UPCA UPCE UPC_SUPPLEMENTAL_2DIGIT UPC_SUPPLEMENTAL_5DIGIT EAN13 EAN8 Interleaved2of5 Standard2of5 Industrial2of5 CODE39 CODE39Extended CODE39_Mod43 ...

Obr. 30: Metody a data třídy Barcode

Knihovna je zpracována pro velké množství typů čárových kódů, což přidává variabilitu navrhovaných štítků. Nicméně existuje možnost potřeby použít čárový kód, který má specifické vlastnosti a byl například vyvinut přímo firmou, která potřebuje navrhovat štítky. Tento požadavek je možné korigovat vytvořením speciální třídy, která bude obsluhovat pouze potřeby tohoto kódu.

Na Obr. 31 jsou viditelné možnosti nastavení čárových kódů. Pokud by došlo k nesprávnému nastavení parametrů, je zobrazena hláška s obsahem problému. Mezi tyto problémy může patřit například nedostatečné nastavení velikosti pro čárový kód. Velikost lze nastavit téměř libovolně, nicméně vnitřní algoritmus knihovny hlídá velikosti bodů a tím pádem dochází ke skokové změně velikosti až při přesáhnutí hraniční velikosti.

Rozvržení dialogu je podobné jako u třídy Text. Ve spodní části je potvrzení nastavených parametrů a nad ním je náhled. Náhled také zobrazuje modře celkovou velikost štítku pro správné nastavení velikosti čárového kódu. Při volbě variabilního obsahu se kód vykreslí modře a je zpřístupněna volba tagu. Oproti textu je zde volba pro zobrazení zakódovaného textu pod čárovým kódem.



Obr. 31: Dialog nastavení pro čárové kódy

6.1.4 Dvoudimenzionální kódy

Pro generování dvoudimenzionálních kódů byla využita knihovna pod názvem ZXing. Knihovna je volně dostupná pod licencí Apache Licence 2.0. Je tedy modifikovatelná a volně šiřitelná. Tato knihovna podporuje větší množství kódů jak dvoudimenzionálních, tak čárových, nicméně její relativní složitost vedla k rozhodnutí využít ji pouze na generování základních dvoudimenzionálních kódů. Za základní dvoudimenzionální kódy byly zvoleny Data Matrix, PDF417 a QR kód. Volba odpovídá kódům popsáných v kapitole 2. Výčet metod a data třídy jsou zobrazeny na Obr. 32.

2DCode	2D Code Data	2DCode Type
-&2DLibrary:.netRefnum -&Encoding options:.netRefnum -2D Code Data:2D Code Data <hr/> +Create() +Data to 2DCode() +Data To String() +Destroy() +Draw Black() +Draw Blue() +Draw() +Exeption Error Info Dialog() +Init() +Load() +Replace Variable Content() +Save() +Settings Dialog()	+2DCode Type:2DCode Type +Content:string +Variable Content:boolean +Tag:string	DataMatrix PDF417 QR Code

Obr. 32: Metody a data třídy 2DCode

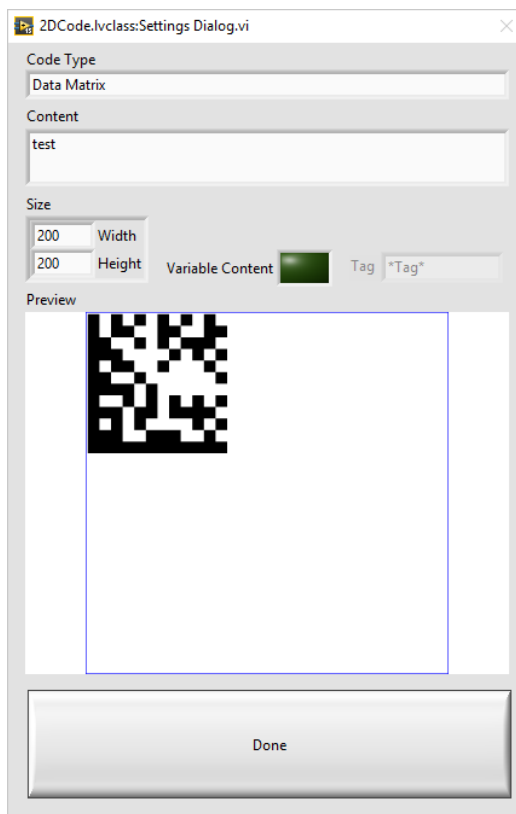
Při použití knihovny se definuje typ dvoudimenzionálního kódu pro třídu, která obstarává samotné generování kódu. Poté pro svou funkci potřebuje další třídu, která obstarává nastavení parametrů pro jednotlivé typy kódů. Pro využití pokročilých nastavení by bylo tedy nutné využít nějakou formu dynamického dialogu. Tento dialog by umožnil nastavení specifických parametrů pro jednotlivé kódy. Implementace takového dialogu by byla velice náročná.

Proto bylo přistoupeno k zpracování pouze základních společných nastavení a specifické vlastnosti jsou ponechány na výchozích hodnotách. V budoucnu je plánováno zpracování dynamického dialogu a použití této knihovny pro práci i s čárovými kódy. Omezení množství použitých knihoven třetích stran je výhodné pro správu softwaru a možnosti jeho testování z hlediska stability i výkonu.

Problém nastal při požadavku na vykreslení kódu v jiné barvě. Pro označení variabilních dat, tedy ve změně kódu v náhledu na modrou. Knihovna nenabízí snadnou volbu barvy. Proto se využívá schopnosti .NET Frameworku využít při vykreslení matici pro úpravu barev. Tato matice byla nastavena, aby upravila černou barvu na modrou.

Dialog nastavení poskytuje volbu potřebných parametrů. I u dvoudimenzionálních kódů si knihovna spravuje nastavené velikosti a změna vykreslovaného obrazu se projeví až při přesahu hraničních hodnot. Poskytuje také nastavení pro zvolení variabilního

obsahu, který bude doplněn až před konečným překladem a také umožňuje volbu tagu pro nahrazení. Dialog nastavení je podobný ostatním a je vidět na Obr. 33.



Obr. 33: Dialog nastavení pro dvoudimenzionální kódy

6.1.5 Obrazová data

Bitmap je název třídy, která byla vytvořena pro práci s obrazovými daty. Pro použití na štítku je možné načíst data ze souborů typu bmp, jpg, gif a png. Lze využít i barevné obrázky a loga. Barvy jsou převedeny do stupňů šedi pomocí matice barev. Strukturu třídy Bitmap je možné vidět na Obr. 34.

Bitmap	Bitmap Data	Rotate
-&Original Bitmap:.netRefnum -&ModBitmap:.netRefnum -&Image Attributes:.netRefnum -&Color Matrix:.netRefnum -Bitmap Data:Bitmap Data <hr/> +Create() +Data to Picture() +Data To String() +Destroy() +Draw() +Init() +Load() +Open Image File() +Picture to Data() +Save() +Settings Dialog()	+Threshold:float +Rotate:Enum +Data:string	Rotate180FlipXY,RotateNoneFlipNone Rotate90FlipNone,Rotate270FlipXY Rotate180FlipNone,RotateNoneFlipXY Rotate270FlipNone,Rotate90FlipXY RotateNoneFlipX,Rotate180FlipY Rotate90FlipX,Rotate270FlipY Rotate180FlipX,RotateNoneFlipY Rotate270FlipX,Rotate90FlipY

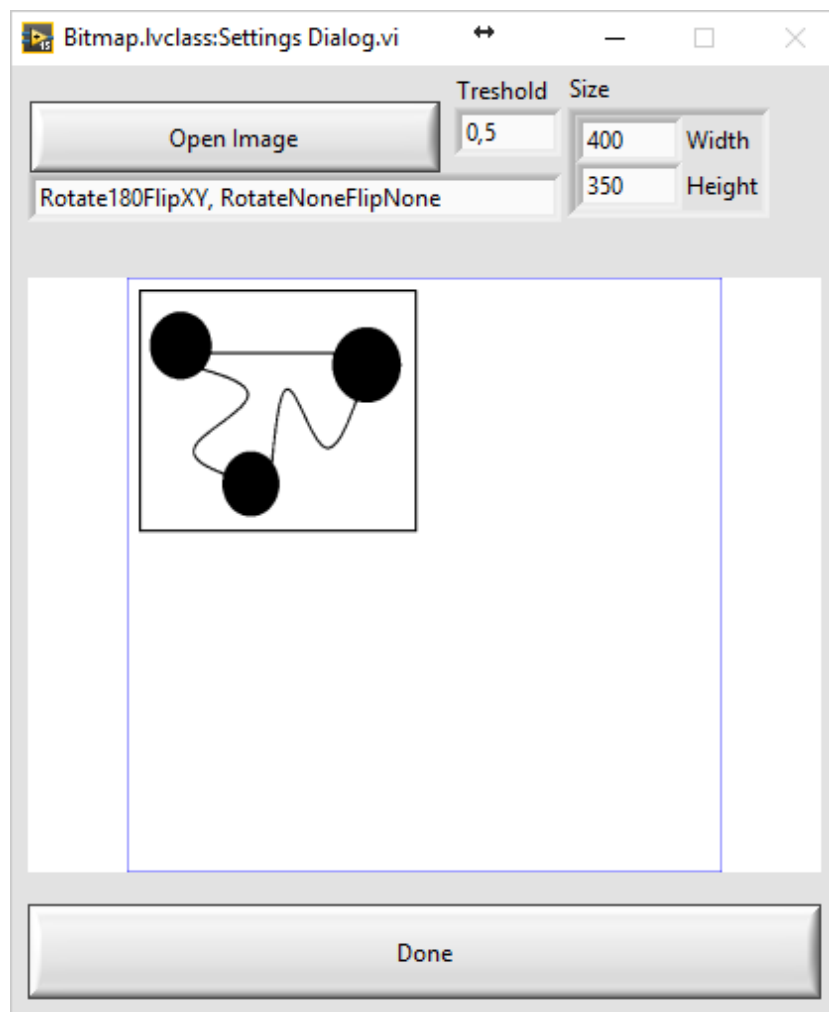
Obr. 34: Metody a data třídy Bitmap

Při požadavku na použití obrazových dat je zobrazen dialog pro výběr souboru, který je následně načten a poté zobrazen v dialogu nastavení. Dialog pro nastavení umožňuje dodatečnou změnu obrázku, který je zobrazen v náhledovém okně, ve kterém je modře zobrazena celková velikost štítku.

Lze upravit míru prahování obrázku ze stupňů šedi, která je důležitá pro nastavení odstínu, který bude zvolen jako práh pro změnu na černou. Samotný převod barev na stupně šedi využívají matice pro úpravu barev.

Vždy se uchovává originální obraz ve stupních šedi s originálním rozlišením. Tento obrázek se ukládá do souboru štítku, aby byl štítek přenositelný na jiné počítače bez nutnosti kontroly přítomnosti souboru. Pro ukládání do souborů se využívá mechanika .NET Frameworku pro vytvoření memory streamu a jeho následné uložení. Do memory streamu se ukládá obrázek formátovaný jako PNG. Nicméně se ukládají pouze data bez hlavičky souboru.

Lze zvolit i rotaci a otočení obrázku kolem jednotlivých os. Jsou dostupné různé kombinace pro dosažení požadovaného efektu.



Obr. 35: Dialog nastavení třídy Bitmap

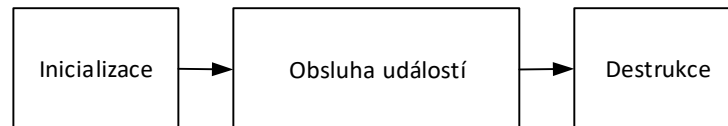
6.2 Část Editor

Pro návrh štítku a práci s již vytvořenými štítky byla vytvořena třída, která má tuto funkcionalitu. Třída se nazývá ZEdit. Využívá jednotlivé prvky z obsahové části a umožňuje tak vytvoření typového štítku. Protože zajišťuje rozsáhlejší funkcionalitu, je tato třída obsáhlejší než předchozí v obsahové části. Jak je možné pozorovat na Obr. 36 obsahuje třída velké množství referencí na pracovní součásti editoru.

ZEdit	Editor Data	Size
-Editor Data:Editor Data -Content:Prototype[] -&Preview:.netRefnum -&Image:.netRefnum -&Graphics:.netRefnum -&Events:Event Callback Refnum[] -&EventReg:Event Registration Refnum -&EventsReg:User Event Refnum -&MCListBox:Multicolumn Listbox Refnum -Events:Events ----- +&MCListBox Get() +Application() +Clear Content() +Content Get() +Content Length Get() +Content position Write() +Create() +Delete Content() +Destroy() +Draw Editor area() +Edit Content() +Event Destroy() +Event Init() +Image and Graphics Init() +Init() +Label Settings Dialog() +Load Editor Data() +Load() +MCL Position Change() +Mouse Down Callback() +Mouse Leave Callback() +Mouse Move Callback() +Mouse Up Callback() +Position Ctrl Position Change() +Render and Highlight() +Save Editor Data() +Save() +Select Dialog() +Size Dialog() +Size Get() +Transform XY coordinates() +UI MCListbox Init() +UI Picture Init() +Ui Init() +Update MCLB()	+Size:Size +Resolution:Enum +Field Origin X +Field Origin Y +Label Rotation:Enum	-Width:int -Height:int ----- Label Rotation Rotate 90 Rotate 180 Rotate 270 ----- Resolution 230 dpi 300 dpi 600 dpi

Obr. 36: Metody a data třídy ZEdit

Při spuštění hlavního podprogramu nazvaného Application dojde k přípravě a zobrazení editoru. Editor se nadále řídí pomocí jednoduchého schématu, které je vidět na Obr. 37. Po počáteční inicializaci s výchozími hodnotami dojde k čekání na další událost. Všechny další uživatelské vstupy obsluhují právě pomocí událostí.

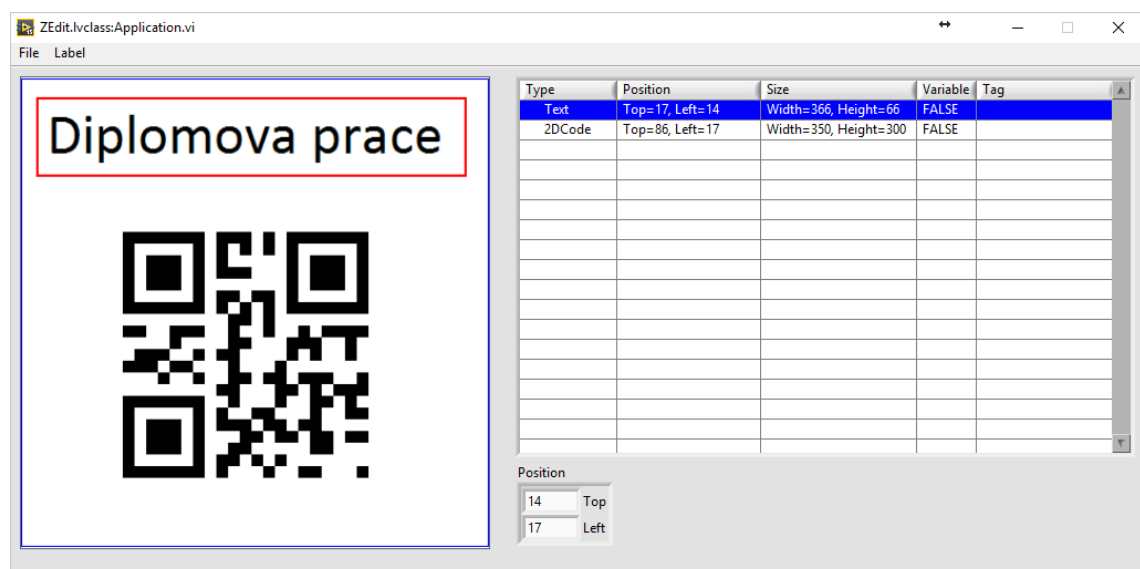


Obr. 37: Jednoduché schéma funkce editoru

Editor dovoľuje meniť nastavenie štítka, a to ako jeho veľkosti, tak voľbu rozlíšenia tiskárny. Rozlíšenie ovplyvňuje pomer tisknutelných bodov na milimetr, pričom editor považuje jeden pixel za jeden tisknutelný bod. Základní jednotkou, s ktorou pracuje editor, je pixel (tisknutelný bod).

Okno editoru se skládá z náhledu, který dominuje v levé části a poté seznamu obsahu v části pravé. Pod seznamem obsahu je také přítomné editovatelné zobrazení pozice právě vybraného obsahu. Toto rozvržení je viditelné na Obr. 38. Pomocí menu lze štítek vygenerovat do xml souboru pro využití v překladači. Dále lze v menu zvolit volbu načtení štítku pro další editaci, volbu nového štítku a také ukončení editoru.

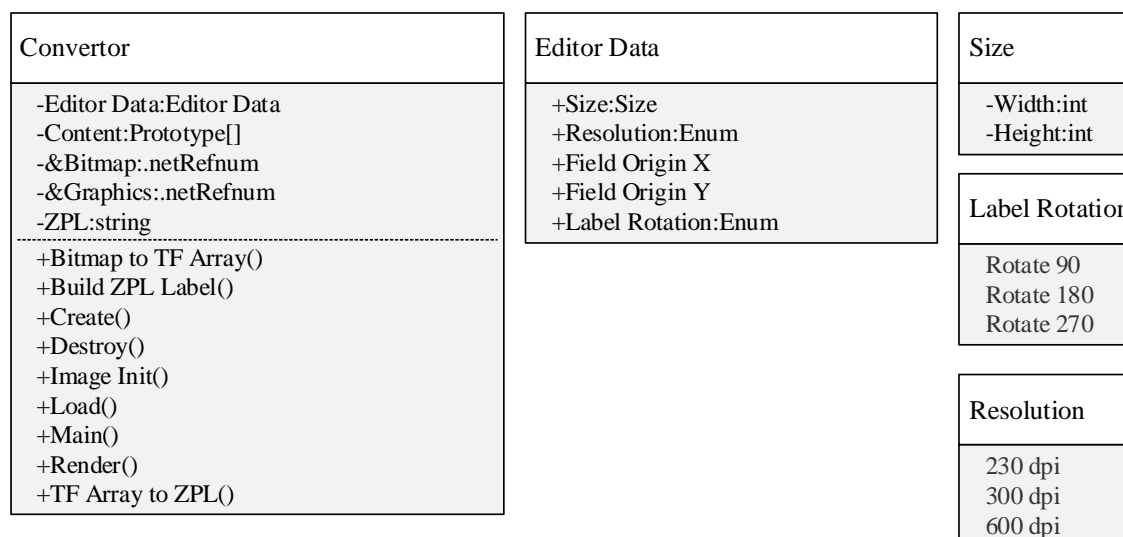
Práce s editorem spočívá v přidání nového obsahu pomocí kontextové nabídky nad seznamem obsahu. Tato volba vyvolá dialog pro zvolení typu obsahu a poté jeho nastavovací dialog. Po nastavení a potvrzení je obsah vložen do hlavního náhledu a lze mu určit pozici. Zvolenou část obsahu lze přemísťovat pomocí myši a pro přesnou pozici je dostupné již zmíněné editační pole. Samozřejmostí je opětovná editace vlastností obsahu a také odebrání obsahu ze štítku. V seznamu obsahu jsou zobrazeny základní informace, jako je typ obsahu, jeho velikost a údaje o variabilním obsahu.



Obr. 38: Ukázka hlavního okna editoru

6.3 Preprocesor a překladač do ZPL II

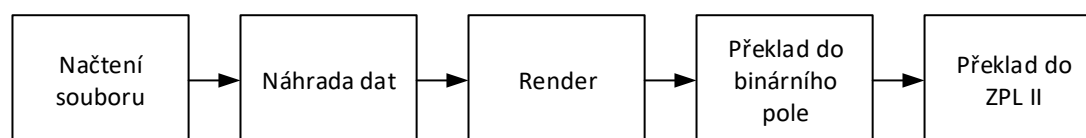
Záměrem třídy Convertor je přeměna načteného štítku z xml souboru na grafická data v jazyku ZPL II. Metody a data, které k tomuto procesu třída potřebuje, jsou zobrazeny na diagramu na Obr. 39.



Obr. 39: Metody a data třídy Convertor

Přímý převod do jazyka ZPL je sice možný, nicméně je poměrně složitý a popíral by samotnou podstatu důvodu tvorby softwaru. Proto se v načteném štítku nahradí variabilní data a poté se obsah vykreslí do obrázku.

Tento obrázek je nutno přeložit do binární podoby. Binární podobou je myšleno to, že každý bod obrázku je vyjádřen hodnotou pravda nebo nepravda. Takto vytvořené dvourozměrné pole lze poté snadno přeložit do jazyka ZPL II. Při překladu se vloží i ostatní data jako je posun štítku a jeho natočení. Pro vyjádření bitmapy v jazyku ZPL II se používá příkaz ^GF, jeho konfigurace lze nalézt v manuálu. Princip nahrazení je zobrazen na Obr. 40. [20]

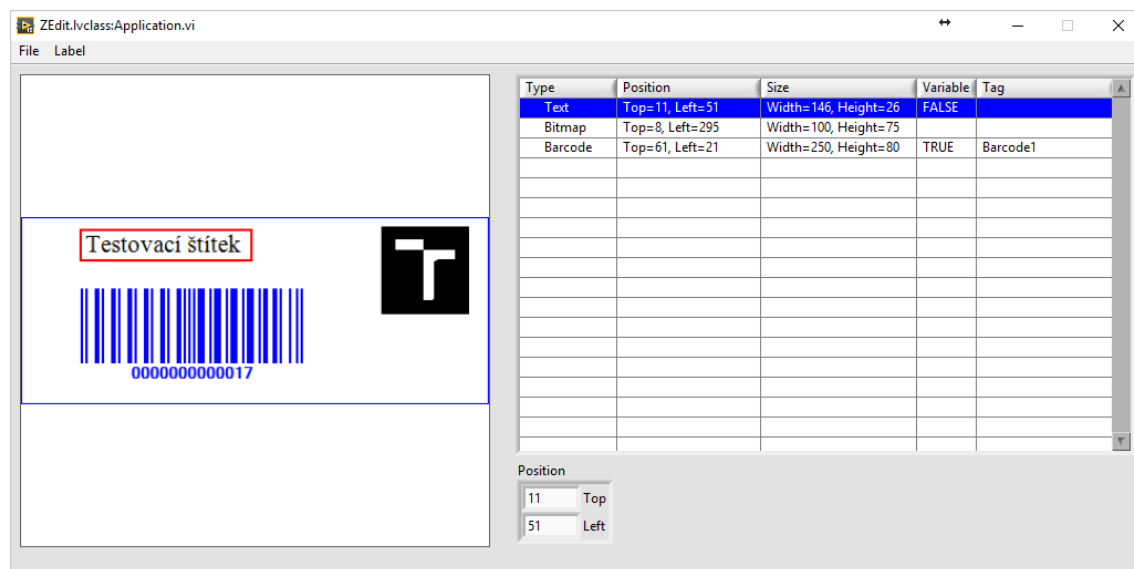


Obr. 40: Postup překladu štítku do ZPL II

Třída Convertor a jeho hlavní metoda Main je nachystaná pro zabudování do vyšších testovacích systémů. Požaduje tedy vstup ve formě cesty k souboru štítku a také data pro nahrazení v případě využití variabilního obsahu. Výstupem je štítek ve formě textu, který lze následně odeslat do tiskárny.

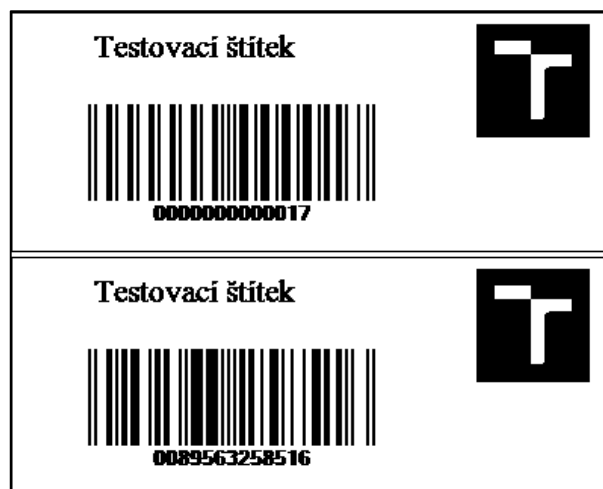
6.4 Vlastnosti realizovaného editoru

Pro zhodnocení aplikace byl vytvořen štítek, který obsahuje název, grafiku v podobě loga a čárový kód reprezentující identifikační číslo. Pro čárový kód byl nastaven příznak variabilních dat a nastaven tag. Při průběhu hlavní metodou třídy Runtime se nahrazují data pro výsledný překlad. Jako výsledná data byly vybrány dvě identifikační čísla, jedno, které je jedna a druhé náhodné. Návrh štítku je viditelný na Obr. 41.



Obr. 41: Návrh štítku pro zobrazení

Pro zobrazení výsledného štítku je obecně nejjednodušší použití online překladače. Například stránka labelary.com nabízí vykreslení dat ze ZPL II do obrázku. Podobnou funkcionalitu obsahují i některé klasické tiskárny firmy Zebra Technologies. Přeložené štítky i s doplněnými daty jsou zobrazeny na Obr. 42. Snížená kvalita odpovídá nastavenému nižšímu rozlišení tiskárny (203dpi)



Obr. 42: Výsledné štítky s různými čísly vykreslené pomocí labelary.com

Navrhovaný štítek ve formátu xml má jednoduchou strukturu, jejíž ukázka je zobrazena na Příloha 1. Na ukázce je viditelný formát zápisu do xml souboru, kdy na začátku jsou uvedena data editoru. Poté následují data jednotlivých tříd obsahu, které jsou podobná pro všechny obsah. Výsledný štítek v tomto formátu má velikost 133 kB. Tato vysoká velikost je způsobena uložením originální grafiky loga v původním rozlišení.

Po překladu dojde k vytvoření textu, který obsahuje štítek v ZPL II. Při zápisu grafiky do ZPL II existuje jistá forma základní komprimace dat. Ukázka přeloženého štítku je viditelná na Příloha 2, ukázka obsahuje pouze část souboru. Výsledná data pro tento štítek mají velikost přibližně 11 kB.

Zlepšením úspory velikosti výsledných dat je použití komprimačního algoritmu, který v základu dokáže využít ZebraDesigner. Nicméně tento algoritmus není popsán a jeho použití bude možné až po jeho prolomení.

Editor je použitelný a nabízí funkcionalitu pro tvorbu a překlad štítků. Další rozšíření jsou možná jak z hlediska obsahu, tak funkcí.

6.5 Další vývoj editoru

Další prací na editoru je propojení překladače štítků přímo do nabídky editoru, aby bylo možné jednoduše jedním kliknutím vytisknout navrhovaný štítek. S touto funkcionalitou souvisí další rozšíření, a to o implementaci správy tiskáren. To znamená, zpracování komunikace s tiskárnou, a to jak po Ethernetu, tak pro sériové lince. Problematické je použití rozhraní USB, pro které se automaticky nainstalují ovladače tisku, které vyžadují přístup přes GDI Windows.

Pro použití na tiskárnách i jiných výrobců lze při požadavku dodělat překladové třídy dle preferencí zákazníků. Také lze v případě potřeby a požadavků rozšířit i obsahovou část softwaru pro další typy obsahu.

Dalším rozšířením je plánované předělání do dynamických dialogů, kdy jednotlivé součásti štítku budou jednotlivá okna. Tento způsob má několik výhod, označování kliknutím, snadnější změna velikosti a další.

Při průběhu práce na návrh editoru byl řešen problém, kdy zákazník měnil tiskové hlavy. Nové tiskové hlavy měly nižší rozlišení a z toho důvodu netiskly štítky správně. Bylo nutné štítky ručně převzorkovat a došlo ke snížení kvality tisknuté grafiky. Tento problém má několik potenciálních řešení. Prvním řešením je pracovat s vyšším rozlišením a následně ho převzorkovat na rozlišení nižší. Při těchto operacích ale mohou vzniknout artefakty na generovaných čárových a dvoudimenzionálních kódech a tím může dojít k způsobení jejich nečitelnosti. Druhým je zavedení základní jednotky milimetr. Při použití na grafická data a na text toto řešení bude fungovat, problém vzniká při požadavku na generování čárových a dvoudimenzionálních kódů. Knihovny pro generování pracují s pixely. Může tak vzniknout požadavek na velikost kódu ve zlomcích pixelu, což není přípustné. Nicméně toto řešení bude s kontrolou a zaokrouhlováním velikostí pixelů použitelné.

7 ZÁVĚR

Práce byla vytvořena z důvodu potřeby jednoduchého softwaru na tvorbu a tisk typových štítků. Navrhovaný editor je koncipován pro zástavbu do stávajících řídicích systémů výrobních linek. Z hlediska obsahu byla práce rozdělena na dvě části, rešeršní část a část praktickou obsahující realizaci.

Rešeršní část popisuje základní prvky obsahu štítků. Vzhledem k absenci normy pro tvorbu typových štítků byly vybrány nejčastěji pozorovatelné součásti typových štítků. Následuje stručný vhled do metod tisku, které se využívají v průmyslu a mohou být využitelné pro tisk typových štítků. Jsou zde zahrnuty různé druhy tisku, z nichž nejdůležitější je termotisk.

Na začátku praktické části jsou uvedeny důvody návrhu softwaru a také koncept jeho rozložení. Poté je popsán konkrétní návrh jednotlivých součástí softwaru a jsou popsány jednotlivé třídy, které těmto částem odpovídají.

Navržený editor umožňuje návrh typových štítků a jejich ukládání do xml souboru. Tyto soubory jsou pak přenosné mezi jednotlivými počítači. Štítek se skládá z různých forem obsahu. Lze nastavovat příznaky na záměnu dat, a takto označená data pak při překladu nahradit konkrétními daty. Tato záměna probíhá i s překladem do jazyka ZPL II v části softwaru, která byla pro tuto funkci vytvořena. Překlad probíhá ve dvou krocích, a to vykreslení do obrazových dat a následného překladu do příkazu jazyka ZPL II pro tisk grafiky. To zaručuje kompletní kontrolu nad tiskem.

Další vývoj je předpokládán v implementaci správy tiskáren do překladové části softwaru. Tato funkce je potřebná pro snadný tisk náhledových štítků, a zlepšení celkové ergonomie návrhu štítků. Další zlepšení návrhu je plánováno v implementaci dynamických dialogů, které umožní uživatelsky přívětivější návrh štítků. Vzhledem ke koncepci celého návrhu softwaru, lze snadno rozšířit funkcionalita jako je například překlad do jiných tiskových jazyků.

8 SEZNAM POUŽITÉ LITERATURY

- [1] PERIASWAMY, S. C. G., D. R. THOMPSON a JIA DI. Fingerprinting RFID Tags. *IEEE Transactions on Dependable and Secure Computing* [online]. 2011, **8**(6), 938-943 [cit. 2017-03-15]. DOI: 10.1109/TDSC.2010.56. ISSN 15455971. Dostupné z: <http://ieeexplore.ieee.org/document/5611545/>
- [2] *Muzeum starých strojů - Historické okamžiky* [online]. b.r. [cit. 2017-03-14]. Dostupné z: <http://www.starestroje.cz/fotoalbum.php?adresar=/prodej/nahradni.dily-stitky&image=19>
- [3] KATO, Hiroko, Douglas. CHAI a Keng T. TAN. *Barcodes for mobile devices*. New York: Cambridge University Press, 2010. ISBN 978-0-521-88839-4.
- [4] GS1 AISBL. *Lineární čárové kódy*. 2015. Dostupné také z: <http://www.gs1cz.org/download/publikace/publikace-linearni-carove-kody.pdf>
- [5] KNUCHEL, Thomas, Tobias KUNTNER, Eva Carolina PATAKI a Andrea BACK. 2D-Codes. *Business*. 2011, **3**(1), 45-48. DOI: 10.1007/s12599-010-0139-z. ISSN 1867-0202. Dostupné také z: <http://link.springer.com/10.1007/s12599-010-0139-z>
- [6] CE marking. *European Commission* [online]. b.r. [cit. 2017-04-11]. Dostupné z: https://ec.europa.eu/growth/single-market/ce-marking_en
- [7] How to identify a Conformité Européenne mark vs a China Export mark. SIRA, ASHWIN. *StarFish Medical* [online]. 2017 [cit. 2017-04-11]. Dostupné z: <http://starfishmedical.com/2017/01/02/conformite-europeenne-mark-vs-a-china-export-mark/>
- [8] ZEBRA TECHNOLOGIES. Zebra™ IQ Color. b.r. Dostupné také z: <https://www.zebra.com/content/dam/zebra/product-information/en-gb/brochures-datasheets/supplies-accessories/iqcolor-en-gb.pdf>
- [9] ZIH CORP. Supplies selector guide. 2017. Dostupné také z: <https://www.zebra.com/content/dam/zebra/product-information/en-us/brochures-datasheets/supplies-accessories/supplies-selector-guide-en-us.pdf>
- [10] H.R.G. Tiskové techniky. H.R.G. *Polygrafické taháky* [online]. b.r. [cit. 2017-04-12]. Dostupné z: http://www.polygraficketahaky.cz/tiskove_techniky
- [11] IZDEBSKA, Joanna. a Sabu. THOMAS. *Printing on polymers: fundamentals and applications*. Iran: Elsevier, 2016. PDL handbook series. ISBN 9780323374682. Dostupné také z: https://app.knovel.com/web/toc.v/cid:kpPPFA0001/viewerType:toc/root_slug:printing-polymers-fundamentals/url_slug:kt010QJIN1

- [12] KREBS, Frederik C. Fabrication and processing of polymer solar cells: A review of printing and coating techniques. *Solar Energy Materials and Solar Cells* [online]. 2009, **93**(4), 394-412 [cit. 2017-04-12]. DOI: 10.1016/j.solmat.2008.10.004. ISSN 09270248. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0927024808003486>
- [13] LEONARDO TECHNOLOGY. Princip termotransferových tiskáren [online]. b.r. [cit. 2017-04-18]. Dostupné z: <http://www.lt.cz/e-learning/termotransfer-technologie/princip-termotransferovych-tiskaren>
- [14] SPI LASERS LIMITED. Fiber Laser Marking of *Plastics* [online]. SPI LASERS LIMITED. b.r. [cit. 2017-04-19]. Dostupné z: <http://www.spilasers.com/application-marking/fiber-laser-marking-of-plastics/>
- [15] SPI LASERS LIMITED. Metal Marking with a *Fiber Laser* [online]. SPI LASERS LIMITED. b.r. [cit. 2017-04-20]. Dostupné z: <http://www.spilasers.com/application-marking/metal-marking-with-a-fiber-laser/>
- [16] Linear & 2D Codes. Millennium signatures [online]. b.r. [cit. 2017-04-25]. Dostupné z: <http://millenniumsinglesmarkingsystems.com/products/linear-2d-codes/>
- [17] Zebra Technologies Corporation History [online]. b.r. [cit. 2017-04-26]. Dostupné z: <http://www.fundinguniverse.com/company-histories/zebra-technologies-corporation-history/>
- [18] ZEBRA. ZT 400. ZEBRA. ZT400 SERIES INDUSTRIAL PRINTERS [online]. b.r. [cit. 2017-04-18]. Dostupné z: <https://www.zebra.com/gb/en/products/printers/industrial/zt400-series.html>
- [19] CAB. ZPL Emulation Manual. 1. 2013. Dostupné také z: <https://www.cab.de/media/pushfile.cfm?file=2114>
- [20] ZEBRA. Zebra Programming Guide. 2005. Dostupné také z: <https://www.zebra.com/content/dam/zebra/manuals/en-us/software/zplii-pm-vol1.pdf>
- [21] ZEBRA TECHNOLOGIES. ZebraDesign Pro: User Guide. 2011. Dostupné také z: <https://www.zebra.com/content/dam/zebra/manuals/en-us/software/zebradesigner-pro-ug-en.pdf>
- [22] CODESOFT 4.2 and above: Convert a *Label File from another labeling program* [online]. 2007 [cit. 2017-05-15]. Dostupné z: <http://www.qualityserviceandsupport.com/brady/article/1318>
- [23] INBOX - obrázky. INBOX [online]. b.r. [cit. 2017-05-13]. Dostupné z: <http://www.inbox.szm.com/pictures.htm>

9 SEZNAM OBRÁZKŮ

Obr. 1: Historický typový štítek [2].....	17
Obr. 2: Ukázka EAN-13	18
Obr. 3: Ukázka Code 128	19
Obr. 4: Ukázka Code 39	20
Obr. 5: Ukázka struktury Data Matrix [3]	21
Obr. 6: Segmenty QR kódu [3]	22
Obr. 7: Ukázka PDF417.....	23
Obr. 8: Ukázka CE značek [7]	24
Obr. 9: Ukázka samolepek pro tisk [9]	25
Obr. 10: Popis principu sítotisku [10].....	26
Obr. 11: Schéma hlavy pro tvoření souvislého proudu kapek	27
Obr. 12: Schéma hlavy založené na teplotním principu	27
Obr. 13: Hlava s piezokrystalem.....	28
Obr. 14: Elektrostatická hlava	28
Obr. 15: Akustická hlava	28
Obr. 16: Princip tamponového tisku [10]	29
Obr. 17: Shéma přímého termotisku [13]	30
Obr. 18: Schéma hlavy pro termtransferový tisk [13]	30
Obr. 19: Různé formy laserového značení [14].....	31
Obr. 20: Ukázka Data Matrix kódu vytvořeného pomocí mikroúderů [16]	32
Obr. 21: Ukázka moderní tiskárny Zebra ZT400 [18].....	33
Obr. 22: Základní pracovní okno programu	36
Obr. 23: Ukázka správného (vlevo) a špatného tisku (vpravo)	38
Obr. 24: Nastavovací dialog program INBOX [23]	40
Obr. 25: Ukázka jednoduchého dialogu.	41
Obr. 26::Schéma obsahové části	44
Obr. 27: Metody a data třídy Prototype	45
Obr. 28: Metody a data třídy Text	46
Obr. 29: Dialog nastavení textu v levé části i s ukázkou fontů v pravé části	47
Obr. 30: Metody a data třídy Barcode	47
Obr. 31: Dialog nastavení pro čárové kódy	48
Obr. 32: Metody a data třídy 2DCode	49
Obr. 33: Dialog nastavení pro dvoudimenzionální kódy	50
Obr. 34: Metody a data třídy Bitmap	50
Obr. 35: Dialog nastavení třídy Bitmap.....	51
Obr. 36: Metody a data třídy ZEdit.....	52
Obr. 37: Jednoduché schéma funkce Editoru	53
Obr. 38: Ukázka hlavního okna editoru.....	53
Obr. 39: Metody a data třídy Convertor	54

Obr. 40: Postup překladač štítku do ZPL II	54
Obr. 41: Návrh štítku pro zobrazení.....	55
Obr. 42: Výsledné štítky s různými čísly vykreslené pomocí labelary.com	55

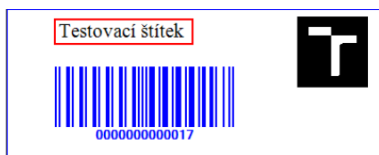
10 SEZNAM PŘÍLOH

CD

Příloha 1: Ukázka části xml souboru štítku i s náhledem (nahore)

Příloha 2: Ukázka přeloženého štítku do ZPL II i s náhledem (nahore)

PŘÍLOHY



```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<ZE DataVersion="1.0.0" FileVersion="1.0.0">

  <ZE_Editor Name="Editor">
    <Cluster Name="Editor Data">
      <Cluster Name="Size">
        <Item Name="Width" Value="400"/>
        <Item Name="Height" Value="160"/>
      </Cluster>
      <Item Name="Resolution" Value="203 dpi"/>
      <Item Name="Field Origin X" Value="0"/>
      <Item Name="Field Origin Y" Value="0"/>
      <Item Name="Label Rotation" Value="No Rotation"/>
    </Cluster>
  </ZE_Editor>

  <ZEContent Name="Text">
    <Cluster Name="Prototype Data">
      <Cluster Name="Size">
        <Item Name="Width" Value="146"/>
        <Item Name="Height" Value="26"/>
      </Cluster>
      <Cluster Name="Position">
        <Item Name="Top" Value="11"/>
        <Item Name="Left" Value="51"/>
      </Cluster>
      <Item Name="Type" Value="Text"/>
    </Cluster>
    <Cluster Name="TextData">
      <Item Name="Font Size" Value="16"/>
      <Item Name="Font" Value="Times New Roman"/>
      <Item Name="Text" Value="Testovací štítek"/>
      <Item Name="Variable Content" Value="FALSE"/>
      <Item Name="Tag" Value="*Tag*"/>
      <Item Name="Orientation" Value="4"/>
    </Cluster>
  </ZEContent>
.
.

```

Příloha 1: Ukázka části xml souboru štítku i s náhledem (nahore)

[illegible]

Príloha 2: Ukážka preloženého štítku do ZPL II i s náhľadom (nahore)